



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**DEVELOPMENT OF A BIG DATA APPLICATION
ARCHITECTURE FOR NAVY MANPOWER,
PERSONNEL, TRAINING, AND EDUCATION**

by

Khristian C. Caindoy
Armin Moazzami
Anthony M. Santos

March 2016

Thesis Advisor:
Second Reader:

Magdi Kamel
Albert Barreto

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2016		3. REPORT TYPE AND DATES COVERED Master's thesis
4. TITLE AND SUBTITLE DEVELOPMENT OF A BIG DATA APPLICATION ARCHITECTURE FOR NAVY MANPOWER, PERSONNEL, TRAINING, AND EDUCATION			5. FUNDING NUMBERS	
6. AUTHOR(S) Khristian C. Caindoy, Armin Moazzami, and Anthony M. Santos				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number NPS.2016.0039-IR-EP5-A.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) Navy Manpower, Personnel, Training, and Education (MPTE) decision makers require improved access to the information obtained from the vast amounts of data contained in a number of disparate databases/data stores in order to make informed decisions and understand second- and third-order effects of those decisions. Toward this end, the research effort of this thesis was two-fold. First, this thesis examined and proposed an end-to-end application architecture for performing analytics for Navy. Second, it developed a decision tree model to predict retention of post-command aviators, using the Cross-Industry Standard Process for Data Mining (CRISP-DM), in support of one Navy MPTE's concerns: retention in post-command aviator community. This research concluded that with the exponential collection and growth of diverse data, there is a need for a combination of Big Data and traditional data warehousing architectures to support analytics at MPTE. The data-mining effort developed a preliminary predictive model for post-command aviation retention and concluded that the number of NOBCs, particularly non-aviation NOBCs, was the most important indicator for predicting retention. Additional data sources particularly those that contain Fitness Reports/Evaluations need to be included in order to improve the accuracy of the model.				
14. SUBJECT TERMS Big Data, application architecture, enterprise architecture, OPNAV N1, manpower, personnel, training, education, predictive modeling, CRISP-DM, aviation community retention			15. NUMBER OF PAGES 135	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**DEVELOPMENT OF A BIG DATA APPLICATION ARCHITECTURE FOR
NAVY MANPOWER, PERSONNEL, TRAINING, AND EDUCATION**

Khristian C. Caindoy
Lieutenant, United States Navy
B.A., University of Washington, 2010

Armin Moazzami
Lieutenant, United States Navy
B.S., University of North Carolina at Charlotte, 2006

Anthony M. Santos
Lieutenant, United States Navy
B.S., University of San Diego, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN NETWORK OPERATIONS AND TECHNOLOGY

from the

**NAVAL POSTGRADUATE SCHOOL
March 2016**

Approved by: Magdi Kamel, Ph.D.
Thesis Advisor

Albert Barreto
Second Reader

Dan Boger, Ph.D.
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Navy Manpower, Personnel, Training, and Education (MPTE) decision makers require improved access to the information obtained from the vast amounts of data contained in a number of disparate databases/data stores in order to make informed decisions and understand second- and third-order effects of those decisions. Toward this end, the research effort of this thesis was two-fold. First, this thesis examined and proposed an end-to-end application architecture for performing analytics for Navy. Second, it developed a decision tree model to predict retention of post-command aviators, using the Cross-Industry Standard Process for Data Mining (CRISP-DM), in support of one Navy MPTE's concerns: retention in post-command aviator community.

This research concluded that with the exponential collection and growth of diverse data, there is a need for a combination of Big Data and traditional data warehousing architectures to support analytics at MPTE. The data-mining effort developed a preliminary predictive model for post-command aviation retention and concluded that the number of NOBCs, particularly non-aviation NOBCs, was the most important indicator for predicting retention. Additional data sources particularly those that contain Fitness Reports/Evaluations need to be included in order to improve the accuracy of the model.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	PROBLEM STATEMENT	1
B.	PURPOSE STATEMENT.....	2
C.	RESEARCH QUESTIONS.....	2
D.	RESEARCH METHODS.....	3
E.	POTENTIAL BENEFITS OF RESEARCH	3
F.	THESIS ORGANIZATION.....	3
II.	NAVY MANPOWER, PERSONNEL, TRAINING, AND EDUCATION	5
A.	MPTE OVERVIEW	5
1.	Bureau of Naval Personnel.....	6
2.	Navy Personnel Command.....	6
3.	Navy Education and Training Command	7
4.	Navy Manpower Analysis Center.....	7
B.	TALENT MANAGEMENT INITIATIVE	7
C.	CURRENT SYSTEMS AND PROCESSES AT MPTE AND THEIR LIMITATIONS	9
D.	MPTE CHALLENGES	10
E.	CHAPTER SUMMARY.....	12
III.	DATA AND APPLICATION ARCHITECTURE.....	13
A.	DATA WAREHOUSES	13
1.	Enterprise Data Warehouse.....	14
2.	Operational Data Store.....	16
3.	Data Vault Architecture	17
4.	Hub and Spoke Data Marts Architecture.....	18
B.	BIG DATA ARCHITECTURE	19
1.	Data Sources Layer.....	20
a.	<i>Structured Data.....</i>	<i>21</i>
b.	<i>Semi-Structured and Unstructured Data</i>	<i>22</i>
c.	<i>MPTE Data Sources</i>	<i>23</i>
2.	Ingestion Layer.....	23
a.	<i>Validation and Cleansing</i>	<i>24</i>
b.	<i>Transforming and Compressing.....</i>	<i>25</i>
c.	<i>Ingestion Options</i>	<i>25</i>
3.	Storage Layer	28
a.	<i>NoSQL Databases</i>	<i>29</i>

	<i>b. Hadoop Distributed File System.....</i>	<i>32</i>
4.	Hadoop Infrastructure Layer	36
5.	Hadoop Platform Management Layer Analytics	39
	<i>a. MapReduce.....</i>	<i>39</i>
	<i>b. Pig.....</i>	<i>40</i>
	<i>c. Hive.....</i>	<i>41</i>
	<i>d. Impala.....</i>	<i>42</i>
	<i>e. Mahout</i>	<i>42</i>
	<i>f. Zookeeper</i>	<i>43</i>
6.	Analytics Engine	44
	<i>a. Data at Rest</i>	<i>45</i>
	<i>b. Streaming Data</i>	<i>45</i>
7.	Visualization Layer	46
8.	Security and Monitoring Layer	48
9.	Hadoop Distributions	49
	<i>a. Cloudera Distribution of Hadoop.....</i>	<i>50</i>
	<i>b. IBM InfoSphere BigInsights.....</i>	<i>50</i>
	<i>c. Amazon Elastic MapReduce.....</i>	<i>51</i>
C.	CHAPTER SUMMARY	51
IV.	CROSS-INDUSTRY STANDARD PROCESS FOR DATA MINING.....	53
A.	CRISP-DM OVERVIEW	53
B.	CRISP-DM PHASES.....	54
1.	Business Understanding	55
	<i>a. Determine Business Objectives.....</i>	<i>55</i>
	<i>b. Assess the Situation.....</i>	<i>56</i>
	<i>c. Determine Data Mining Goals</i>	<i>56</i>
	<i>d. Produce Project Plan</i>	<i>56</i>
2.	Data Understanding.....	56
	<i>a. Collect Initial Data.....</i>	<i>57</i>
	<i>b. Describe Data</i>	<i>57</i>
	<i>c. Explore Data</i>	<i>57</i>
	<i>d. Verify Data Quality</i>	<i>57</i>
3.	Data Preparation.....	57
	<i>a. Select and Clean Data.....</i>	<i>58</i>
	<i>b. Construct, Integrate, and Format Data</i>	<i>58</i>
4.	Modeling	58
	<i>a. Select Modeling Technique</i>	<i>58</i>
	<i>b. Generate Test Design.....</i>	<i>59</i>
	<i>c. Build and Assess Model.....</i>	<i>59</i>

5.	Evaluation	59
a.	<i>Evaluate Results</i>	59
b.	<i>Review Process</i>	59
c.	<i>Determine Next Steps</i>	60
6.	Deployment	60
a.	<i>Plan for Deployment, Monitoring, and Maintenance</i>	60
b.	<i>Produce Final Report and Review Project</i>	60
C.	CHAPTER SUMMARY	61
V.	APPLICATION OF CRISP-DM TO THE POST-COMMAND AVIATOR COMMUNITY	63
A.	BUSINESS UNDERSTANDING	63
1.	Business Objectives	63
2.	Assess Situation	64
3.	Determine Data Mining Goals	64
4.	Produce Project Plan	64
B.	DATA UNDERSTANDING	65
1.	Data Collection	65
2.	Data Description	66
3.	Data Exploration	67
4.	Data Quality	70
C.	DATA PREPARATION	72
1.	Converting Timestamp Data Types to Integers	72
2.	Calculating Difference in Years	74
3.	Calculating Total Deployment Time	74
4.	Converting Fields to Flags	76
5.	Code Mapping	77
6.	Aggregation and Count of Attributes	78
7.	Defining Target Variable through Binning	81
8.	Understanding New Variables	82
D.	MODELING	91
1.	Selecting the Model	91
2.	Generating Test Design	92
3.	Building the Model	92
4.	Assessing the Model	95
E.	EVALUATION	96
1.	Evaluating the Results	96
2.	Reviewing the Process	97
3.	Determining the Next Steps	97
F.	DEPLOYMENT	97

G.	CHAPTER SUMMARY.....	97
VI.	SUMMARY, CONCLUSION, AND RECOMMENDATIONS	99
A.	SUMMARY	99
B.	REVIEW OF RESEARCH QUESTIONS.....	100
1.	Data Architecture Questions.....	101
2.	Data Mining Project Questions.....	103
C.	RECOMMENDATIONS.....	104
1.	Other Internal Databases and Fitness Reports/Enlisted Evaluations	104
2.	External Data Sources	104
3.	Security and Privacy	105
D.	CONCLUSION	105
	LIST OF REFERENCES	107
	INITIAL DISTRIBUTION LIST	113

LIST OF FIGURES

Figure 1.	Typical Data Warehouse Architecture.....	15
Figure 2.	Operational Data Store Architecture.....	17
Figure 3.	Data Vault Architecture	18
Figure 4.	Hub and Spoke Data Marts Architecture	18
Figure 5.	The Big Data Architecture	20
Figure 6.	Variety of Data Sources	21
Figure 7.	Example of Structured Data.....	22
Figure 8.	Examples of Unstructured Data	22
Figure 9.	Ingestion Layer Components	23
Figure 10.	Sqoop Architecture	27
Figure 11.	Flume Agent.....	28
Figure 12.	NoSQL Databases.....	31
Figure 13.	HBase Layout.....	32
Figure 14.	HDFS Architecture	34
Figure 15.	DataNode Replication	35
Figure 16.	Cluster Size Growth Projection	38
Figure 17.	Typical Big Data Hardware Topology.....	39
Figure 18.	MapReduce Task	40
Figure 19.	Zookeeper Topology	44
Figure 20.	Data Analytic Workflow	44
Figure 21.	Physical Architecture of Apache Storm.....	46
Figure 22.	Spark versus MapReduce.....	46
Figure 23.	Cloudera Distribution of Hadoop.....	50
Figure 24.	Amazon EMR Interaction with Other Cloud Services	51
Figure 25.	CRISP-DM Process Model	55
Figure 26.	Extracted NMPBS Files	65
Figure 27.	Distribution of Aviation Designators.....	67
Figure 28.	Percentage of Ranks.....	68
Figure 29.	Distribution of Males and Females	68
Figure 30.	Distribution of Separation Codes.....	69

Figure 31.	Distribution of Advanced Qualification Designation Codes	69
Figure 32.	Distribution of Subspecialty Codes	70
Figure 33.	Distribution of the Number of Dependents per Household	71
Figure 34.	Distribution of the Deployment Duration	72
Figure 35.	Year Eligible to Retire Formula.....	73
Figure 36.	Year Retired Formula	73
Figure 37.	Replacing Blank Values Expression.....	75
Figure 38.	Deployment Duration Expression.....	75
Figure 39.	Convert Flag Fields Example.....	76
Figure 40.	Map Codes Expression	77
Figure 41.	Aggregation of Subspecialty Code Fields.....	78
Figure 42.	Mapping of NOBCs to a Category.....	79
Figure 43.	NOBC Aggregation and Count.....	79
Figure 44.	Aggregation and Count of Total NOBCs.....	80
Figure 45.	Deriving Non-Aviation NOBCs	80
Figure 46.	Creating Bin Values.....	81
Figure 47.	Deriving Target Variable	82
Figure 48.	Distribution of IA/GSA Assigned with Target Overlay	83
Figure 49.	Distribution of Joint Qualification with Target Overlay.....	83
Figure 50.	Distribution of Joint Service Officer Tour with Target Overlay	84
Figure 51.	Distribution of JPME I Only Complete with Target Overlay.....	84
Figure 52.	Distribution of JPME I and II Complete with Target Overlay	85
Figure 53.	Distribution of Master’s Degree with Target Overlay	85
Figure 54.	Distribution of War College Education with Target Overlay	86
Figure 55.	Distribution of Language Skills with Target Overlay.....	86
Figure 56.	Distribution of Source Code with Target Overlay	87
Figure 57.	Distribution of Count of Subspecialty Codes with Target Overlay	87
Figure 58.	Distribution of Deployment Duration with Target Overlay	88
Figure 59.	Distribution of Count of AQDs with Target Overlay	88
Figure 60.	Distribution of Count of Total NOBCs with Target Overlay	89
Figure 61.	Distribution of Count of Aviation NOBCs with Target Overlay.....	90
Figure 62.	Distribution of Count of Non-Aviation NOBCs with Target Overlay	90

Figure 63.	QUEST Model Decision Tree.....	94
Figure 64.	C & R Decision Tree.....	95
Figure 65.	Model Summary.....	95
Figure 66.	Model Accuracy.....	96
Figure 67.	Predictor Importance.....	96

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	Retrieved Fields from NMPBS.....	66
Table 2.	Fields with Complete Values.....	70
Table 3.	Fields Converted to Integers.....	74
Table 4.	Expressions Used to Calculate Difference in Years.....	74
Table 5.	Fields Converted to Flags Names.....	76
Table 6.	Code Mapping Fields.....	77
Table 7.	Input and Target Variables.....	91

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

API	application program interface
AQD	Additional Qualification Designation
BASE	basically available, soft-state, eventually consistent
BI	business intelligence
BOL	BUPERS Online
BUPERS	Bureau of Naval Personnel
CAP	consistency, availability, and partition tolerance
CDH	Cloudera Distribution of Hadoop
CNO	Chief of Naval Operations
CNP	Chief of Naval Personnel
C&R	Classification and Regression
CRISP-DM	Cross-Industry Standard Process Model for Data Mining
DBMS	database management system
DFAS	Defense Finance and Accounting Service
DOD	Department of Defense
DON	Department of the Navy
EC2	Elastic Compute Cloud
EDVR	Enlisted Distribution Verification Reports
EMR	Elastic MapReduce
ESR	Electronic Service Record
ETL	extraction, transformation, and loading
FITREP	Fitness Report
FLTMPS	Fleet Training Management Planning System
GSA	Global War on Terror Support Assignment
HDFS	Hadoop Distributed File System
IA	Individual Augmentee
IS	information science
IT	information technology
JBOD	just a bunch of disks
JDBC	java database connectivity

JPME	Joint Professional Military Education
JSO	Joint Service Officer
JVM	java virtual machine
MPP	massively parallel processing
MPTE	Manpower, Personnel, Training, and Education
NAVMAC	Navy Manpower Analysis Center
NES	Navy Enlisted System
NETC	Naval Education and Training Command
NIC	network interface card
NMPBS	Navy Manpower Programming Budget System
NOBC	Navy Officer Billet Classification
NoSQL	Not only Structured Query Language
NPC	Navy Personnel Command
NROTC	Naval Reserve Officer Training Corps
NSIPS	Navy Standard Integrated Personnel System
NTMPS	Navy Training Management Planning System
ODC	Officer Data Cards
ODCR	Officer Distribution Control Reports
OLAP	Online Analytical Processing
OPINS	Officer Personnel Information Systems
OPNAV	Office of the Chief of Naval Operations
PII	personally identifiable information
QUEST	Quick, Unbiased, Efficient Statistical Tree
RDBMS	relational database management system
S3	Simple Storage Service
SECNAV	Secretary of the Navy
SQL	Structured Query Language
SOA	service-oriented architecture
USNA	United States Naval Academy
USNI	United States Naval Institute

ACKNOWLEDGMENTS

We would like to thank Professor Magdi Kamel, Buddy Barreto, and Alida Laney for their outstanding support in this thesis. Professor Kamel, thank you for your guidance and patience with us throughout this entire research. Buddy, thank you for your technical expertise and letting us use your lab; sorry that we could not get your cluster up and running. Alida, thank you for all your wonderful insights and help in getting this research going.

We especially would like to thank our families for all their unwavering love and support.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The amount of data collected and stored is growing exponentially. Challenges associated with large databases are not only the volume of available data but also the variety of the data types, the velocity at which the data arrives, and the ability to analyze and extract value from such data. Current database technologies are unable to keep up with the volume of data being created and the demand of the users requiring the data. It is expected that there will be 4300% increase in annual data generation by 2020 with 1/3 of that data living in or passing through the cloud (CSC, 2012). This rapid growth of data can be attributed to the switch from analog to digital technologies and the dramatic rise of unstructured data like photos, videos, and social media (CSC, 2012). Sources of Big Data are unlimited and can come not only from social media, but also from digital devices such as mobile phones, web logs, sensors, etc. The Large Hadron Collider/Particle Physics is one example of Big Data; it produced 13–15 petabytes of data in 2010 (Kaisler, Armour, Espinosa, & Money, 2013). Logs of web searches or retail transactions are other examples of Big Data (Jacobs, 2009). “What makes data big is repeated observations over time and/or space” (Jacobs, 2009, p. 40). For example, web pages can log millions of visits a day; cell phone databases store time and location every 15 seconds for each of a few millions phones (Jacobs, 2009). Data production does not appear to be stopping anytime soon.

A. PROBLEM STATEMENT

In order to keep up with the rapid growth of data collection, Navy Manpower, Personnel, Training, and Education (MPTE) wishes to harness the power of Big Data to assist in making personnel decisions and provide timely Human Resources information to Sailors; however, it is essential to select the right Big Data architecture suitable to meet such goal. Navy MPTE decision makers require improved access to information obtained from the vast amounts of data contained in a number of disparate databases/data stores in order to make informed decisions and understand second and third order effects of those decisions.

B. PURPOSE STATEMENT

This research will consist of two parts. First, we will examine and propose an end-to-end application architecture for performing analytics for Navy MPTE, looking at both a traditional architecture based on a data warehousing approach and a Big Data architecture. Second, we will perform a predictive analysis on a subset of data focusing on the retention issue of the post-command aviator community.

C. RESEARCH QUESTIONS

The following are the questions posed for this research:

- What are the substantive issues that a Navy MPTE Common Operating Picture is trying to solve?
- What are the various internal and external data sets that need to be analyzed?
- How is the ingestion of the data into the Hadoop environment accomplished from the data sources?
- What are the necessary Hadoop infrastructure hardware and software components needed?
- What are the different types of NoSQL databases that are most suitable to store Navy MPTE data?
- Does Navy MPTE need Big Data technology or should it instead use a robust, high-performing, relational database management system (RDBMS) and traditional Data Warehouse technology?

The following set of questions are for our analysis on the data set of post-command aviators. Answering these questions will assist MPTE leadership in improving the retention rate of post-command aviators.

- What aviation talent is being lost?
- What are some indicators that would lead an Officer to leave the service?
- Can a model be developed from available data to predict post-command aviator retention?

D. RESEARCH METHODS

This research will follow two methods: an enterprise application architecture framework and the Cross-Industry Standard Process for Data Mining (CRISP-DM). An enterprise architecture framework is used by organizations as a guide to align business processes and goals with information systems. It is important to gain a good understanding of each component of an enterprise architecture in order to meet the demands of the organization. The CRISP-DM process will be used to conduct a predictive analysis on the issue of retention for the post-command aviator community for Navy MPTE.

E. POTENTIAL BENEFITS OF RESEARCH

The potential benefits of this research will provide better solutions for improved access to Navy MPTE information that cannot be extracted in its current data form. It will also provide unique insights and tailor human resource services to Navy Leadership and decision makers to improve retention for post-command aviator community.

F. THESIS ORGANIZATION

Chapter II describes Navy MPTE and their organizations and the current challenges of talent management and retention.

Chapter III examines and contrasts a traditional data warehouse and a Big Data Application architecture.

Chapter IV gains an understanding of the CRISP-DM process.

Chapter V describes how the CRISP-DM process was applied to conduct a predictive analysis on retention for the post-command aviator community.

Chapter VI provides a summary of the research, conclusions, and recommendations for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. NAVY MANPOWER, PERSONNEL, TRAINING, AND EDUCATION

Prior to developing an enterprise application architecture or an analytics solution from organizational data, it is important to gain a good understanding of organizational processes and goals. The purpose of this chapter is to gain an understanding of Navy Manpower, Personnel, Training, and Education (MPTE) by examining its organizations, current systems, and processes. We then look into the Talent Management Initiative, a major initiative of MPTE, and how it is meant to improve retention. Lastly, we describe MPTE's challenges with retention.

A. MPTE OVERVIEW

The Chief of Naval Personnel (CNP) is responsible to the Chief of Naval Operations (CNO) for Navy's manpower readiness (Department of the Navy [DON], n.d.-a). Dual-titled as the Deputy Chief of Naval Operations (Manpower, Personnel, Training Education/OPNAV N1), the CNP oversees the Bureau of Naval Personnel (BUPERS), Navy Personnel Command (NPC), Naval Education and Training Command (NETC), and the Navy Manpower Analysis Center. Combined, these organizations create and implement the overall strategy and policies concerning manpower and training programs (DON, n.d.-a).

The overall mission of the CNP is to “anticipate Navy warfighting needs, identify associated personnel capabilities, and recruit, develop, manage, and deploy those capabilities in an agile, cost-effective manner” (Hall, 2006, para. 8). In order to achieve this mission, CNP has set forth three strategic objectives for the MPTE domain: Responsive Force Management, Effective Personnel Readiness, and Sound Organizational Alignment (Navy Personnel Command [NPC], 2013). Responsive Force Management focuses on bettering distribution, training, recruiting, and retention in order to meet Fleet manpower requirements (NPC, 2013). Effective Personnel Readiness focuses on developing a ready Sailor through proper training and education and supported by services and resources they and their families need (NPC, 2013). Sound

Organizational Alignment focuses on ensuring the decisions and actions made by OPNAV N1 are aligned with the needs of the fleet (NPC, 2013). It is through these strategic objectives that MPTE will continually support the CNO's principles of Warfighting First, Operate Forward, and Be Ready of Warfighting First, Operate Forward, and Be Ready (NPC, 2013).

1. Bureau of Naval Personnel

The Bureau of Naval Personnel (BUPERS) can be equated to a human resource department of an organization; it provides administrative and policy planning for personnel matters for the Navy. BUPERS was established to advise the CNO on “personnel plans and policies for recruitment, distribution, advancement, compensation, retention, readiness, retirement, and community management for regular and reserve Navy personnel” (Office of the Chief of Naval Operations [OPNAV], 2012b, p. 3). The organization has many functions, which includes developing manpower requirements for current fleet activities and for new Navy programs and initiatives (OPNAV, 2012b). BUPERS additionally implements policy regarding Navy compensation, pay entitlement, and travel reimbursement (OPNAV, 2012b).

2. Navy Personnel Command

The mission of the Navy Personnel Command (NPC) is to “man the Fleet with ready Sailors; supporting their ability to serve from beginning to end” (NPC, 2013, p. 5). In line with CNP's Strategic Objective of Responsive Force Management and Sound Organizational Alignment, NPC works closely with operational forces to streamline fleet manning processes in delivering highly trained Sailors and ensuring critical billets are filled on time to reach the goal of 90% fit 60 days before deployment (NPC, 2013). Other functions of NPC include establishing programs that promotes diversity and improves quality of life and conducting manpower research to attract new recruits and retain talented Sailors (Bureau of Naval Personnel [BUPERS], 2002).

3. Navy Education and Training Command

Training is imperative in keeping Sailors abreast of current technologies, practices and issues they face on a daily basis, and the Navy Education and Training Command (NETC) is charged with this responsibility. NETC's mission is to "educate and train Sailors and provide those tools and opportunities which enable life-long learning, professional and personal growth and development, and ensuring fleet readiness and mission accomplishment" (OPNAV, 2012a, p. 1). NETC is responsible to the CNO and Commander, U.S. Fleet Forces Command for training and educating the fleet (OPNAV, 2012a).

4. Navy Manpower Analysis Center

The mission of the Navy Manpower Analysis Center (NAVMAC) is to "define, translate, and classify the Navy's work into a workforce structure and position demand signal to sustain a combat ready force" (Navy Manpower Analysis Center [NAVMAC], 2015). The four mission areas of NAVMAC are (NAVMAC, 2015):

1. Navy's Occupational Classification Systems
2. Fleet Manpower Requirements Determination
3. Navy's Manpower Management Program Administration
4. Navy's Manpower Information System Business Requirements

Through these four mission areas, NAVMAC defines the Navy's manpower demand by conducting manpower studies, performing manpower assessments, and managing manpower programs and projects (NAVMAC, 2015). The primary outcomes of NAVMAC are effective job and qualifications management, valid ship/squadron manpower requirements, effective manpower management processes and policies, and effective information system performance of manpower processes (NAVMAC, 2015).

B. TALENT MANAGEMENT INITIATIVE

The Navy provides a worldwide presence, with the objective to be in the right place in the right time. Secretary of the Navy (SECNAV) Ray Mabus describes Sailors as the "greatest edge" (Mabus, 2015). It is the Sailors that allow the Navy to continually

provide worldwide presence. Therefore, recruiting, developing, retaining, and promoting Sailors is critical to the success of the Navy (Mabus, 2015). In a speech at the U.S. Naval Academy (USNA), Mabus (2015, p. 3) stated,

To fight and win in this century we need a force that draws from the broadest talent pools, values health and fitness, attracts and retains innovative thinkers, provides flexible career paths, and prioritizes merit over tenure. Whether we are talking about systems and tactics in the digital age or personnel management, we must evolve to meet the needs of the future battle space and the needs of our people; or we can – we will – lose.

The Navy is shifting its mindset from retaining the most willing to retaining the most talented. It is the most talented Sailors that the Navy is losing to the private sector. During his speech at the USNA, Mabus announced several talent management initiatives to improve retention. One of those initiatives is the establishment of the Office of Talent Optimization at USNA. The purpose of this office is to develop an understanding on the state of the civilian labor market in order to keep the Navy competitive in recruiting the best people (DON, 2015). In association with MPTE policy planners, warfare community leaders, and USNA faculty, the Office of Talent Optimization will work in creating a talented workforce (DON, 2015).

Another talent management initiative is creating an adaptive workforce. Part of creating an adaptive workforce includes opening all operational billets to women (DON, 2015). In order to create an equal opportunity environment for both men and women, Mabus strongly believes in allowing women in positions that were previously not available to them and implementing one standard for both sexes just as long as that standard meets the requirements of the job (DON, 2015). In addition to opening operational billets, Mabus proposed an increased in bonus opportunities. Rather than granting bonuses across the board, DON leaders have the ability to grant them based on specific skill-sets and talent (DON, 2015).

C. CURRENT SYSTEMS AND PROCESSES AT MPTE AND THEIR LIMITATIONS

MPTE uses a number of systems and associated databases to manage its personnel. These include the Officer Personnel Information System (OPINS), the Navy Enlisted System (NES), the Navy Training Management and Planning System (NTMPS), the Navy Standard Integrated Personnel System (NSIPS), and the Navy Manpower Programming and Budget System (NMPBS). In this section we overview each of these systems.

Two authoritative personnel databases for the Navy are the Officer Personnel Information System (OPINS) and Navy Enlisted System (NES). OPINS stores personnel data for active and reserve officers, officer candidates, and midshipmen from the United States Naval Academy (USNA) and Naval Reserve Officer Training Corps (NROTC), and NES for active and reserve enlisted personnel (DON, n.d.-c; DON, n.d.-b). MPTE leadership uses OPINS and NES to determine the health of the officer and enlisted workforce, while the Defense Finance Accounting System (DFAS) uses it to establish pay records (DON, n.d.-c; DON, n.d.-b). The data contained in OPINS is used to prepare Officer Data Cards (ODC) for officers and Officer Distribution Control Reports (ODCR) for afloat and shore activities, which can be assessed through BUPERS Online (BOL) (DON, n.d.-c). The enlisted equivalent to the ODCR is the Enlisted Distribution Verification Reports (EDVR) (DON, n.d.-b). Data quality is important for both OPINS and NES as the data is used for promotion boards and determining enlisted distributions (DON, n.d.-c; DON, n.d.-b). Additionally, reports created for congress to gain insights on the officer and enlisted communities are derived from these databases (DON, n.d.-c; DON, n.d.-b).

Navy Training Management and Planning System (NTMPS) is a database for managing training data and provides users with standard or ad hoc reports (PMW 240, 2015b). NTMPS utilizes a data warehouse/operational data store architecture with business intelligence tools (PMW 240, 2015b). Over 30 approved data sources feed into the NTMPS data warehouse/operational data store (PMW 240, 2015b). Additionally, the NMTPS data warehouse/operational data store is the source of data for NTMPS related

applications like Fleet Training Management and Planning System (FLTMPS), Electronic Training Jacket, and NTMPS Afloat Data Mart (PMW 240, 2015b).

The Navy Standard Integrated Personnel System (NSIPS) integrated legacy human resources management systems into a single database for pay and personnel data (PMW 240, 2015a). It is a web-enabled system allowing Sailors to access their Electronic Service Records (ESR), training data, and career development records from anywhere in the world (PMW 240, 2015a). A future goal for NSIPS is the integration of data from OPINS, NES, and other manpower related databases (PMW 240, 2015a).

Another system available to Navy MPTE is Navy Manpower Programming and Budget System (NMPBS). It is a centralized system for personnel, manpower, and financial-related data sponsored by the CNP (Hamilton, 2015). Navy MPTE uses NMPBS as a decision support tool for resource allocation and community/manpower-related management, providing leadership with real-time data and analysis tools for short, immediate, and long-term planning (Hamilton, 2015). NMPBS is the only Navy system to combine daily personnel transactions with pay data to achieve detailed personnel costs (Office of the Chief of Naval Personnel, 2015). Data sources of NMPBS include OPINS, NES, NMTPS, and NSIPS.

The data that is stored on these disparate systems, if integrated properly, could provide insights of which personnel are the best to recruit, who have aptitude to be successful in their rating or designator, and the likelihood to persevere a full Navy career, making every dollar spent on the service member worth it.

D. MPTE CHALLENGES

Many challenges exist in the process of making personnel decisions. Data exists in separate systems that are not connected or integrated. Additionally, there is a lack of standardization across the Navy's Human Resource Systems. This lack of standardization makes it difficult to integrate and analyze data across disparate systems as it may be formatted differently between systems. This leads to a lack of common terminology among the systems. Due to the stove piping of the databases, data is being duplicated in multiple systems. We foresee a Big Data end-to-end application architecture as a possible

solution for integrating data from these diverse databases as well as other external data sources and hence address many of these challenges. This will enable Navy MPTE to establish and adjust recruitment goals, create and adjust training schedules, force shape, make policy adjustments and implementations, and answer congressional and Office of the Secretary of Defense questions.

Retention has been a lingering issue for Navy MPTE. In an interview with the U.S. Naval Institute (USNI), CNP, Vice Admiral Bill Moran, was asked if there were any jobs in the Navy that were historically difficult to fill. His main concern was regarding the change in the airline business as they continue to expand and increase their pilot workforce (LaGrone, 2014). Airlines are trending towards recruiting command-level aviators because of their experience and background vice officers who have only completed their initial obligation of service (LaGrone, 2014). As Moran pointed out, post-command aviators are choosing to retire at their 20-year mark instead of staying in the service and competing for more senior positions (LaGrone, 2014). To address this retention cliff in the aviation community, a bonus has been reestablished to help incentivize post-command aviators to stay around (LaGrone, 2014).

Aviation is not the only community that is seeing a decrease in retention. VADM Moran identified the nuclear community, both in the submarine and surface force, personnel in the Information Technology (IT) domain, and Special Operations Forces (LaGrone, 2014). Due to the demanding lifestyle of working on nuclear plants, especially on carriers, keeping junior officers and experienced enlisted nukes in the service has been a tough challenge (LaGrone, 2014). In regard to IT, the difficulty arises in retaining high-quality personnel as the commercial industry grows (LaGrone, 2014). Special Operations Forces are dealing with the same retention issue as the post-command aviators. According to VADM Moran, senior leaders in the community are starting to retire earlier than expected, and the cause of that is due to the number of deployments and time away from family (LaGrone, 2014).

E. CHAPTER SUMMARY

In this chapter, we gained understanding of MPTE by examining its organizations, processes, systems, and challenges. This will help in the development of a predictive model and is step one of the CRISP-DM process. The next chapter will look into detail two types of data architectures: a data warehouse that is currently used by MPTE and a Big Data Application architecture.

III. DATA AND APPLICATION ARCHITECTURE

After gaining a good understanding of organizational processes and goals, the next step is to align those processes and goals with an enterprise architecture framework. An enterprise architecture framework is meant to guide an organization through the implementation of an IT architecture. The current architecture that Navy MPTE uses is a data warehouse; however, based on the understanding of Navy MPTE business processes and needs, a Big Data Application Architecture is suited to meet their requirements. The purpose of this chapter is to examine and contrast traditional data warehouses and Big Data Application architecture.

A. DATA WAREHOUSES

Typically, data is stored in operational databases where business intelligence (BI) systems can access them directly (Kroenke & Auer, 2014). Operational databases work for smaller databases and simple reporting, but not for larger databases and complex applications (Kroenke & Auer, 2014). Applications can experience a degradation in performance and burden the database management system (DBMS) when data is queried for BI systems.

To address this issue, organizations started developing separate repositories for data analysis and decision support called data warehouses. According to Vaisman and Zimányi (2014), a data warehouse is a collection of integrated data that will be used for analytics. Data warehouses are also defined by four characteristics: subject-oriented, integrated, nonvolatile, and time-varying. Subject-oriented means that the data contained in a data warehouse is relevant to an organization (Vaisman & Zimányi, 2014). Integrated refers to data that is combined from several operational databases (Vaisman & Zimányi, 2014). Nonvolatile means that the data in a data warehouse is neither modified nor removed which expands the lifetime of that data to exceed that of an operational database (Vaisman & Zimányi, 2014). Time-varying means that different values are retained as well as the time that the changes occurred for the same information (Vaisman & Zimányi, 2014).

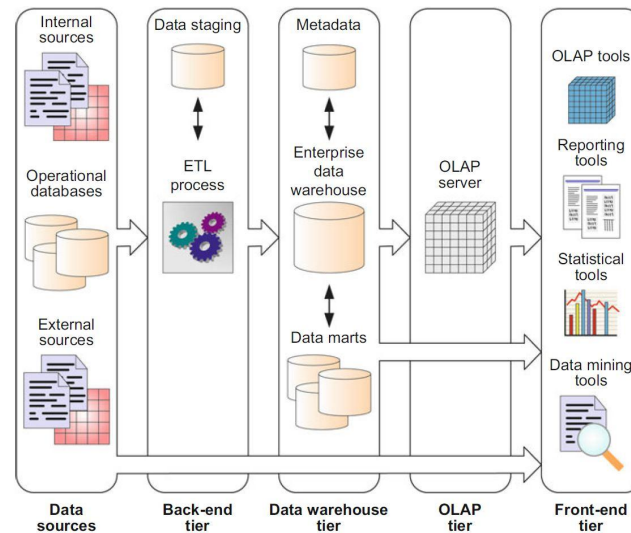
The goal of data warehouses is to analyze the data of an entire organization (Vaisman & Zimányi, 2014). Occasionally, departments within an organization may only need a subset of data within a data warehouse. For example, the supply department would require only logistical data. These specialized data warehouses are called data marts. According to Vaisman and Zimányi (2014), data marts can be shared among the different departments if required by organizational processes.

Data warehouses are developed using either a bottom-up or top-down approach. The bottom-up approach is where data marts are built first then combined to create an enterprise wide data warehouse (Vaisman & Zimányi, 2014). This approach is suited for organizations that need fast results and do not want to risk the time and costs of building an enterprise data warehouse. The top-down approach is the more classic approach where organizations first create an enterprise data warehouse and then derive data marts from it (Vaisman & Zimányi, 2014). In this case, data marts are considered to be a logical view of a data warehouse (Vaisman & Zimányi, 2014).

1. Enterprise Data Warehouse

A typical data warehouse consists of four tiers: back-end tier, data warehouse tier, Online Analytical Processing (OLAP) tier, and the front-end tier (Vaisman & Zimányi, 2014). Figure 1 depicts the architecture for a data warehouse.

Figure 1. Typical Data Warehouse Architecture



Source: Vaisman, A., & Zimányi, E. (2014). Data warehouse concepts. *Data warehouse systems* (pp. 53–87) Berlin: Springer, p. 77.

The back-end tier is where the extraction, transformation, and loading (ETL) process occurs (Vaisman & Zimányi, 2014). During the extraction phase, data is gathered from multiple data sources that are either internal or external to the organization (Vaisman & Zimányi, 2014). In the transformation phase, the data is modified to match the format of the data warehouse (Vaisman & Zimányi, 2014). According to Vaisman and Zimányi (2014), the transformation phase is where data is cleaned to remove errors and inconsistencies and is converted to a standardized format; it is then integrated to combine and summarize data from different sources. Once the transformation phase is complete, the data is loaded into the data warehouse (Vaisman & Zimányi, 2014). Vaisman and Zimányi (2014) added that the loading phase may also include the refreshing of the data warehouse where updated data is loaded to provide near real-time data for analysis. Furthermore, the back-end tier may also include an operational data store to complement the ETL process where data from the sources would undergo successive modifications prior to being loaded into the data warehouse.

The data warehouse tier is composed of a data warehouse or a combination of a data warehouse and data marts. Additionally, this tier also consists of the metadata repository. Vaisman and Zimányi (2014, p. 78) defined metadata as “data about data” and

are classified either as technical and business metadata. Technical metadata describes the structure or representation of the data and how it will be stored and accessed (Vaisman & Zimányi, 2014). The business metadata differs in that instead of describing the physical aspect of data, it describes how the data will be presented, which is done through organizational rules and policies (Vaisman & Zimányi, 2014). For example, the business metadata for ZIP codes may be to represent them as nine digits vice five. The metadata repository in a data warehouse may contain information that defines the ETL process, the data sources, or how the data warehouse or data marts are structured (Vaisman & Zimányi, 2014).

The next tier in the data warehouse architecture is the OLAP tier. It consists of an OLAP server, which provides a wide array of applications such as performance reporting or what-if analysis all of which require historical, projected, and derived data (Vaisman & Zimányi, 2014; Hyperion, 2000). With a combination of standard access tools and an analytic engine, users can gain a deeper understanding of data (Hyperion, 2000).

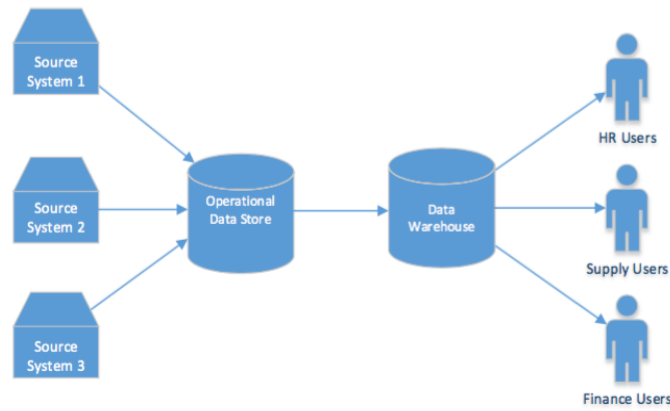
The last tier of the data warehouse architecture is the Front-End tier. This tier contains tools for users to use to manipulate data in the data warehouse (Vaisman & Zimányi, 2014). Some client tools include OLAP tools, reporting tools, statistical tools, and data mining tools. OLAP tools allow users to explore and manipulate warehouse data, and facilitates the formulation of complex queries, called ad hoc queries, that may “involve large amounts of data” (Vaisman & Zimányi, 2014, p. 79). According to Vaisman and Zimányi (2014), reporting tools create, deliver, and manage paper-based or interactive, web-based reports. Furthermore, statistical tools “analyze and visualize cube data using statistical methods” (p. 79). Lastly, data mining tools provide insights into patterns and trends that cannot be derived with standard analytic tools (Vaisman & Zimányi, 2014).

2. Operational Data Store

An operational data store uses the same concept as a data warehouse by integrating data from multiple sources but differs in the fact it can present data in real-time (LeBlanc, Moss, Sarka, & Ryan, 2015). Prior to entering an operational data store,

data goes through the ETL process in order to be cleansed, transformed, and integrated. Figure 2 shows the data architecture of an operational data store with a data warehouse attached. The purpose of attaching a data warehouse is to pull information for historical tracking and reporting. Though operational data stores present a real-time view of data, it does not have the ability to store historical information for an extended period of time (LeBlanc, Moss, Sarka, & Ryan, 2015).

Figure 2. Operational Data Store Architecture

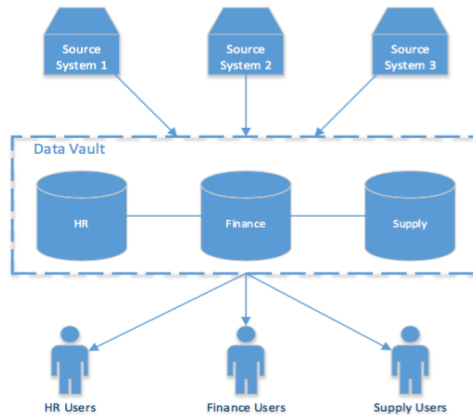


Adapted from LeBlanc, P., Moss, J. M., Sarka, D., & Ryan, D. (2015). *Applied Microsoft business intelligence*. Indianapolis, IN: John Wiley & Sons, Inc.

3. Data Vault Architecture

According to Linstedt (2015), a data vault focuses on tracking historical data within normalized tables, which directly supports different functional areas of organizations. It was designed to adapt quickly to the business environment and is capable of precisely identifying the business needs. This type of data architecture is a combination of a star and third-normal form model (Linstedt, 2015). Figure 3 presents a simple data vault architecture. Benefits of a data vault are the capability to extract data from big data, and continuous assimilation of unstructured data (Linstedt, 2015).

Figure 3. Data Vault Architecture

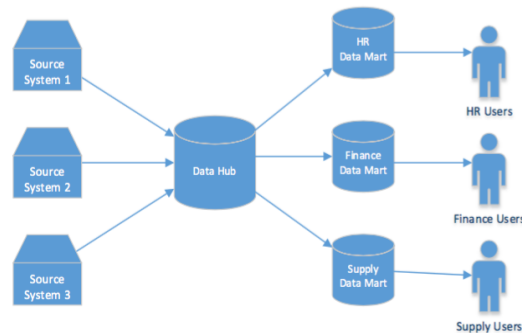


Adapted from LeBlanc, P., Moss, J. M., Sarka, D., & Ryan, D. (2015). *Applied Microsoft business intelligence*. Indianapolis, IN: John Wiley & Sons, Inc.

4. Hub and Spoke Data Marts Architecture

Hub and spoke data mart architecture is a mix approach that combines an enterprise data warehouse and many departmental data marts (LeBlanc, Moss, Sarka, & Ryan, 2015). The hub and spoke data mart, as seen in Figure 4, includes a central data hub that contains all the data of an organization. The data hub then transfers information to the respective data marts. Benefits of hub and spoke data marts include the central storage of information, only allowing users access to information that they need, and can support the development of new data marts in parallel with the existing system (LeBlanc, Moss, Sarka, & Ryan, 2015).

Figure 4. Hub and Spoke Data Marts Architecture



Adapted from LeBlanc, P., Moss, J. M., Sarka, D., & Ryan, D. (2015). *Applied Microsoft business intelligence*. Indianapolis, IN: John Wiley & Sons, Inc.

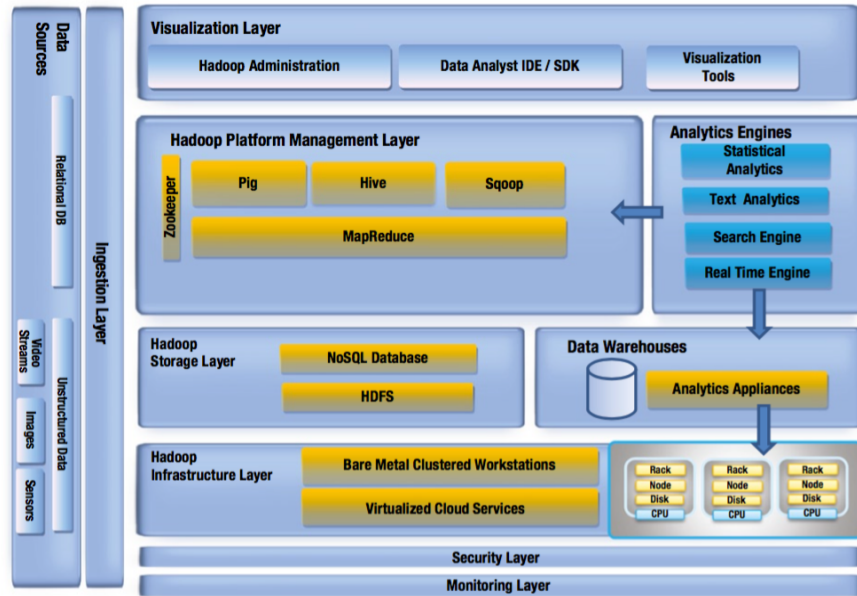
B. BIG DATA ARCHITECTURE

Big Data is a complicated concept that shares the same challenges as the service-oriented architecture (SOA) and cloud computing (Sawant & Shah, 2013). When implementing a Big Data Application Architecture, non-functional requirements such as availability, security, scalability, and performance must be taken into account (Sawant & Shah, 2013). An end-to-end application architecture is required to accurately delineate best practices and guidelines in order to deal with business objectives and non-functional requirements (Sawant & Shah, 2013). A Big Data application architecture has many advantages. One is that it can store Terabytes or Petabytes worth of data. The difference from the other data architectures is that Big Data can support a wide range of data sources. Enterprise data warehouses are limited to structured data while Big Data can support both structured and unstructured.

Figure 5 outlines the essential components of a Big Data Application Architecture. The first component of a Big Data architecture is the data sources. These sources can either be structured, semi-structured, or unstructured data (Sawant & Shah, 2013). Once the sources have been identified, the data passes through the ingestion. The purpose of the ingestion layer is to filter and process large amounts of rapidly changing data from multiple sources (Sawant & Shah, 2013). Following the ingestion, data is stored in the Hadoop Storage Layer or a Data Warehouse. The Hadoop Infrastructure Layer supports the Hadoop Storage Layer and can either be bare metal clustered workstations stored on-site or be virtual and stored in the cloud through a third party (Sawant & Shah, 2013). The Hadoop Platform Management Layer sits on top of the physical infrastructure layer and manages HDFS by utilizing different applications (Sawant & Shah, 2013). Tools that are part of the Hadoop Platform Management Layer are MapReduce, Sqoop, Pig, Hive, Impala, and Zookeeper (Sawant & Shah, 2013). The purpose of the Visualization Layer is to provide data analysts the ability to quickly assess the overall picture of the data in various modes (Sawant & Shah, 2013). When developing a Big Data Application Architecture, it is important to consider security. The Security Layer is where proper authorization and authentication methods are applied to the analytics (Sawant & Shah, 2013). The Monitoring Layer houses the tools needed to

monitor the distributed Hadoop grid architecture (Sawant & Shah, 2013). The following sections provide details of each component.

Figure 5. The Big Data Architecture

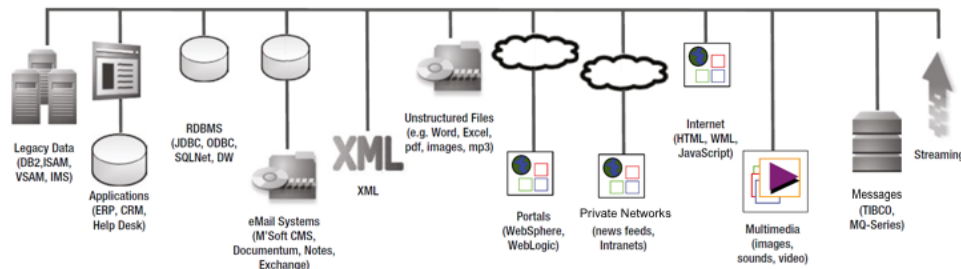


Source: Sawant, N., & Shah, H. (2013). *Big data application architecture Q&A: A problem - solution approach* (1st ed.). Berkeley, CA: Apress, p. 10.

1. Data Sources Layer

The purpose of the data layer of a Big Data Application Architecture is to identify the data needed for analysis and where they would be coming from. Data can come from a variety of sources, whether it is from relational databases or social media. Figure 6 illustrates the different variety of data sources. Data can be further broken down into two types: structured and unstructured data.

Figure 6. Variety of Data Sources



Source: Sawant, N., & Shah, H. (2013). *Big data application architecture Q&A: A problem - solution approach* (1st ed.). Berkeley, CA: Apress, p. 11.

a. **Structured Data**

Structured data is most commonly found in relational databases or spreadsheets, where the data can be organized in columns and rows as seen in Figure 7. This type of data is easily sifted and navigated because each field is designated with a data type, whether it be numerical, binary, or a string. If a proper data model is followed when entering the data, then the structure will remain constant, giving both computers and human users the ability to understand the contents of the structured data. This is because when creating a data model, relationships of the data are determined and also the constraints and specifications of the data when entered.

The common tools used for structured data are relational database management systems (RDBMS) and spreadsheet software; Microsoft Access and Excel are well known examples. These tools simplify creating, entering, editing and analyzing data for use. As the data gets larger, RDBMS software becomes more efficient, mainly because of embedded Structured Query Language (SQL), which is a language, used to manage and query data within the RDBMS. While structured data is what organizations operate on, unstructured data is becoming more abundant and is not yet analyzed in many legacy applications.

Figure 7. Example of Structured Data

	A	B	C	D	E	F	G	H
1	STUDENT ID	FIRST NAME	LAST NAME	DOB	MAJOR	GRADUATION DATE	ADVISOR	THESIS
2	123456	JOHN	DOE	1/1/1980	PHYSICS	3/25/2016	EINSTEIN	STRING THEORY
3	234567	NANCY	DREW	2/3/1985	ECONOMICS	3/25/2016	NASH	FINANCIAL CRISIS
4	987654	BOB	ROBERTS	3/4/1983	HISTORY	3/25/2016	LEWIS	AMERICAN REVOLUTION
5								

b. Semi-Structured and Unstructured Data

The emergence of new technologies and social media has opened the door for the proliferation of semi-structured and unstructured data. Unstructured data refers to data that does not follow a specified format, while semi-structured data refer to data that contain elements that are structured and elements that has no predefined structure. For instance, email messages are considered semi-structured data, as it includes defined fields such as sender, addressee, and date sent; however, the message portion of an email has no predefined structure. It is usually very hard and costly to process and analyze unstructured data. According to Syed, Gillela, and Venugopal (2013), 90 percent of Big Data is unstructured data. As seen in Figure 8, unstructured data can be anything from social media posts, word documents, PDFs, videos, and photos.

Figure 8. Examples of Unstructured Data



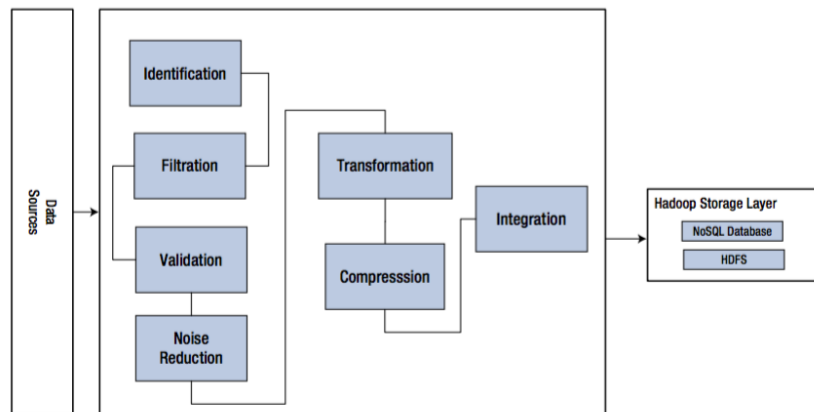
c. *MPTE Data Sources*

The data sources to be examined for this research are NTMPS, NSIPS, and OPINS, which are all relational and thus structured. NMPBS is a data warehouse that integrates data from NTMPS, NSIPS, and OPINS.

2. Ingestion Layer

Once the data for analysis has been identified, it has to be filtered, cleansed, transformed, integrated, and imported into the storage layer. This is the function of the Ingestion Layer where relevant data for analysis is first separated from noise and then loaded into big data repository. According to Sawant and Shah (2013), the signal-to-noise ratio is usually 10:90. Furthermore, the Ingestion Layer should be designed to support the 3 V's (volume, velocity, and variety) of Big Data and be able to authenticate and transform information into the Big Data architecture for analysis. Data can be integrated manually using Hadoop command lines or automated using tools like Sqoop or Flume. Figure 9 illustrates the basic components of the Ingestion Layer.

Figure 9. Ingestion Layer Components



Source: Sawant, N., & Shah, H. (2013). *Big data application architecture Q&A: A problem - solution approach* (1st ed.). Berkeley, CA: Apress, p. 13.

The Identification phase involves identifying the different formats of data, particularly unstructured data (Sawant & Shah, 2013). Data is then filtrated to keep those relevant to the organization. Next, the data goes through a round of validation to ensure it

meets organizational needs. The Noise Reduction phase involves the cleansing of data and the removal of noise (Sawant & Shah, 2013). In the Transformation phase, the data is de-normalized, joined, or summarized (Sawant & Shah, 2013). During the Compression phase, the data size is reduced (Sawant & Shah, 2013). Finally, Integration sends the final data set into the Hadoop Storage Layer.

a. Validation and Cleansing

Part of the Ingestion Layer involves the validation and cleansing of the data. With large data sets, data quality is always of big concern as it can skew or falsify analysis results. Data can suffer from three main types of problems; it can be inconsistent, invalid, or corrupt (Cloudera, 2013). Data is considered inconsistent when it has minor formatting variations. Invalid data is data that is considered incorrect but conforms to the expected format. Corrupt data is data that does not conform to the expected format. The methods used for validation and cleansing are parsing, standardization, abbreviation expansion, and updating missing fields.

Differences and ambiguity in data may confuse individuals and even automated systems. For example, the state of California can be represented in several formats such as CA, Calif., or California. Humans can easily read and understand the different formats but not so much with automated systems. According to Loshin (2012), parsing is the process of identifying and analyzing patterns in data. Data values are compared to defined patterns or regular expressions to distinguish between valid or invalid values.

Standardization takes the parsing process a step further and transforms data values to a standard format. This process can change full words to abbreviations or correct common misspellings. Continuing with the state example, if an application identifies the different representations of California, it can transform all values to a standardized two-letter abbreviation of CA.

Abbreviations is a compact representation of a recognized value and standardizing abbreviations is another aspect of data validation and cleansing (Loshin, 2012). There are several types of abbreviations: ones that shortens words to just the prefix like ST for street and INC for incorporated, and another that removes letters from word like MGR

for manager. Acronyms are another type of abbreviations. To cleanse abbreviations, they must be parsed and transformed to a format according to organizational rules.

Missing fields in data can mean more than people may expect. It may be that there is actually no value for this field, the value is unknown at the time of entry, or the value does not conform to a predefined set of approved values. The missing value may also be the result of errors in the original value. It is important to document and define rules on how to deal with missing values. Simply filling missing values may be counterproductive and will not help with the final analysis.

b. Transforming and Compressing

More than likely, data that is received is not in a format that is required. Sometimes the format given is suitable for data collection but not for analysis. Certain formats scale better than others, offer better performance, and are better for long-term storage. The only solution is to transform the data to the format that is needed.

When converting small data sets, UNIX commands such as `tr`, `join`, `paste`, `sed`, or `awk` can be used to change the format. Some applications like Microsoft Excel can export files to the required format or the “Save As” feature can change the extension of the file. Scripts or small programs can also be written up to convert data.

c. Ingestion Options

There are several options to ingest data into a Big Data architecture. The simplest and sometimes fastest option is a file transfer. Other options are the use of tools such as Sqoop and Flume. Some considerations must be taken when deciding between the three options. If the existing data format is suitable for analysis, then a file transfer is suitable (Grover, Malaska, Seidman, Shapira, 2015). Sometimes errors can occur mid-transfer; therefore, for reliability reasons, Sqoop or Flume are the better options (Grover et al., 2015). If the transformation of data is required then Flume is the correct tool to use (Grover et al., 2015).

(1) File Transfer

Two Hadoop commands are used in the file transfer process: `hadoop fs -put` or `hadoop fs -get` (Grover et al., 2015). The `-put` command imports files into HDFS and the `-get` command exports the files out of HDFS. File transfers are considered as an all-or-nothing approach; if an error occurs during the transfer, no data will be written or read (Grover et al., 2015). By default, file transfers are single-threaded and do not support parallelization (Grover et al., 2015). Transformations of data is not supported by file transfers (Grover et al., 2015). File transfers can support different types of data (i.e., text, binary, or images) because it loads data by bytes (Grover et al., 2015).

When a `-put` job is initiated, the transfer process utilizes either the double-hop or single-hop approach. According to Grover et al. (2015), the double-hop approach is the slower option because there is an additional read/write step on the Hadoop edge node prior to reaching HDFS. In some cases, the double-hop approach may be the only option because the external file source cannot be mounted from the Hadoop cluster (Grover et al., 2015). The single-hop approach requires the source device be mounted onto the Hadoop cluster (Grover et al., 2015). This allows for the `put` command to read directly from the source and write directly onto HDFS. When a `-put` job is completed, the file will be placed into HDFS as one or more blocks with each block replicated across the different DataNodes (Grover et al., 2015).

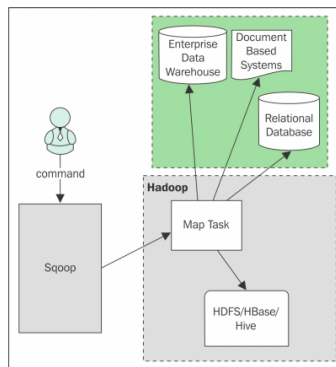
(2) Sqoop

Nearly all structured data is stored in RDBMS and is usually accessed using SQL. It can be used to load flat files into the storage layer; however, this process can be time consuming. A tool called Sqoop can speed up the load process. Sqoop, which is short for SQL to Hadoop, is designed to transfer data between RDBMS, Hadoop, and Not only Structured Query Language (NoSQL) databases (Teller, 2015). Sqoop automates both the import and export process, relying on the database to describe the schema for the data (Apache, 2016b). Additionally, Sqoop runs the export and import process in parallel with MapReduce, providing for fault tolerance (Apache, 2016b). In order to connect with external databases, Sqoop utilizes connectors and drivers. Drivers connect Sqoop to the

external database, whether it is MySQL, Oracle, DB2, or SQL Server (Teller, 2015). Connectors optimize the data transfer by obtaining metadata information from the database (Teller, 2015).

The Sqoop architecture is shown in Figure 10. When a client initiates a Sqoop command, it retrieves the metadata of the tables, columns, and data types according to the connectors and drivers interfaces (Teller, 2015). The import or export process is then translated to a Map-only job program to load data in parallel between the databases and Hadoop (Teller, 2015). Sqoop has its limitations which include security and configurations concerns. Root access is required to install and configure Sqoop (Teller, 2015). Connectors for Sqoop are only java database connectivity (JDBC) based and have to support the serialization format; otherwise, it cannot transfer data (Teller, 2015). These issues have been corrected with the development of Sqoop 2 (Teller, 2015).

Figure 10. Sqoop Architecture



Source: Teller, S. (2015). *Hadoop essentials*. Birmingham, UK: Packt Publishing, p. 354.

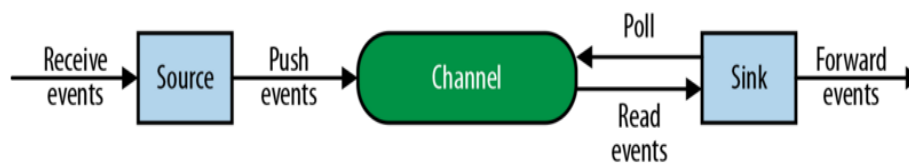
(3) Flume

Flume is another tool used for ingestion. According to Shreedharan (2014), Flume is designed to be a flexible distributed system that is easily scalable and highly customizable. Though mainly used for transferring log data, Flume can be used to move massive amounts of data such as network traffic; social media generated data, or message queue events (Apache, 2012). Some key features of Flume are reliability, recoverable, declarative, and highly customizable (Grover et al., 2015). Flume is reliable because data,

also known as events to Flume, are stored in the channel until delivered to the next stage (Grover et al., 2015). It is recoverable because events can remain on a disk for a period of time and recovered in an event of a failure (Grover et al., 2015). Declarative is meant by no coding is required; the configuration of Flume specifies how components are wired together (Grover et al., 2015). Finally, highly customizable means that the Flume architecture is highly pluggable allowing from the implementation of custom components based on requirements (Grover et al., 2015).

Flume is deployed as agents and consists of three major components as seen in Figure 11. Sources are active components and receive data from external sources (Shreedharan, 2014). They listen to one or more network ports to receive and read data from the local file system and must be connected to at least one channel (Shreedharan, 2014). Channels are the passive component of a Flume agent, acting like a queue, with sources writing to them and sinks reading from them (Shreedharan, 2014). Multiple sources can write to a single channel while sinks can only read from only one channel (Shreedharan, 2014). Sinks continuously communicate with their respective channels to read and remove events then push events onto the next agent or to the storage layer (Shreedharan, 2014).

Figure 11. Flume Agent



Source: Shreedharan, H. (2014). *Using flume*. Sebastopol, CA: O'Reilly Media Inc, p. 25.

3. Storage Layer

The purpose of the Storage Layer is to hold the data needed for analysis. This layer is comprised of NoSQL databases and HDFS. Data can also be stored in a data warehouse; however, the decision where to store data is dependent upon the organization's business processes. If an organization requires real-time analysis of data,

then a NoSQL/HDFS configuration is best solution. However, if an organization requires analysis of a small data set, then a data warehouse is their best option.

a. NoSQL Databases

NoSQL databases non-relational databases where data are not stored in tables (Padhy, Patra, & Satapathy, 2011). The first use of the term NoSQL was in 1998 where relational databases did not use SQL (Strauch & Kriha, 2011). Then in 2009, NoSQL resurfaced at conferences of advocates of non-relational databases who were seeking solutions to problems that relational databases were a bad fit for (Strauch & Kriha, 2011). NoSQL databases started to gain traction because relational databases were becoming slow and expensive, and database administrators wanted to find a more efficient and cheaper way of managing data (Strauch & Kriha, 2011).

Several advantages of NoSQL databases over relational databases include the reading and writing of data quickly, support for mass storage, high scalability, and low cost (Jing, Haihong, Guan, & Jian, 2011). According to Strauch and Kriha (2011), relational databases are highly complex; the variety of features and strict data consistency requirements of the relational databases may be over complicated certain applications. Furthermore, some NoSQL databases provide significantly higher throughput than traditional relational databases. Most NoSQL databases are capable of scaling data horizontally without relying on expensive hardware (Strauch & Kriha, 2011). Unlike with relational databases, machines can be easily added or removed to a NoSQL database with no effect to operations (Strauch & Kriha, 2011). In regards to the speed of reading and writing data, NoSQL databases avoid expensive object-relational mapping (Strauch & Kriha, 2011).

(1) NoSQL Characteristics

The consistency, availability, and partition tolerance (CAP) theorem is widely adopted by the NoSQL community (Strauch & Kriha, 2011). A database is consistent when an update operation is done on the system and all users of the shared data source can see the update (Strauch & Kriha, 2011). Availability is when a database system is designed and implemented in a way to ensure continued operation (Strauch & Kriha,

2011). When a database continues operation when network partitions are present, it is partition tolerant (Strauch & Kriha, 2011). The CAP theorem states that for any system sharing data, it cannot simultaneously have all three properties; therefore, NoSQL databases select two of the three properties (Pokorny, 2013). Many NoSQL databases favor availability and partitioning over consistency, creating new systems known as basically available, soft-state, eventually consistent (BASE) (Padhy et al., 2011).









NoSQL databases scale both vertically and horizontally (Padhy et al., 2011). Traditional relational databases are usually constrained to a single server, scale through adding hardware, and rely on replication to keep databases synchronized (Padhy et al., 2011). According to Padhy et al. (2011), NoSQL databases are designed to operate on one or more servers located physically on-site or in the cloud.

Another characteristic of NoSQL databases is that it has two options of storing data: in-memory or on-disk. According to Padhy et al. (2011), relational databases are traditionally stored on a physical or network drive. Furthermore, data are loaded into memory through a SQL Select script. This process is inefficient as it can take time and require more processing power. NoSQL databases, in order to speed up the process, can be stored in memory then, if necessary, moved onto a disk (Padhy et al., 2011).

(2) Types of NoSQL Databases

The four types of NoSQL databases are key-value stores, column-oriented data stores, document-based data stores, and graph data stores. According to Padhy et al. (2011), data in key-value stores are stored as a key-value pair and can support both structured and unstructured data (Padhy et al., 2011). Furthermore, column-oriented data stores group data by one extendable column. Document-based data stores are where data is stored and organized as a collection of documents (Padhy et al., 2011). On the other hand, graph data stores are designed to effectively manage linked data, and are very effective on application based data (Hecht & Jablonski, 2011). Figure 12 lists several popular NoSQL databases.

Figure 12. NoSQL Databases

Key-Value Data Stores	Column-oriented Data Stores	Document Data Stores	Graph Data Stores
    	     	    	    

Source: Sawant, N., & Shah, H. (2013). *Big data application architecture Q&A: A problem - solution approach* (1st ed.). Berkeley: Apress, p. 14.

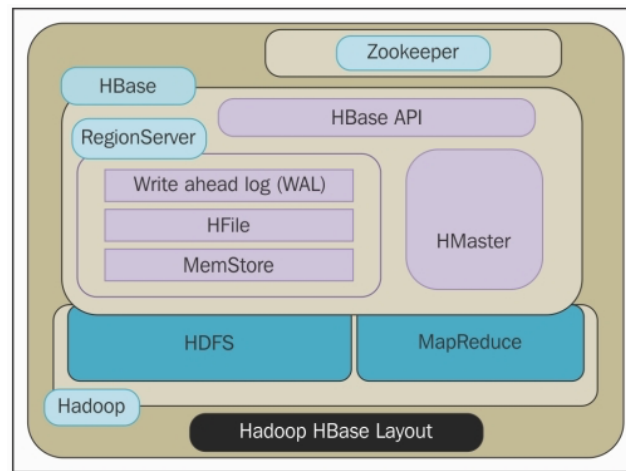
NoSQL databases can be applied in certain business applications. For example, key-value data stores are best for businesses like Amazon where items are paired with a key (Sawant & Shah, 2013). Column-oriented databases are suitable for organizations that analyze web user actions or sensor feeds like social media (Sawant & Shah, 2013). Organizations that require to log and analyze real time data would benefit from document based data stores (Hecht & Jablonski, 2011). Use cases for graph data stores include location based services, knowledge representation, recommendation systems, or any other use that involve complex relationships (Hecht & Jablonski, 2011).

(3) HBase

The main database used by Hadoop is HBase. It is a horizontally scalable, distributed, open source NoSQL database that runs on top of HDFS (Shripurv, 2014). Modeled after Google's BigTable for the Google file system, HBase was developed for Hadoop to support the storage of structural data (Shripurv, 2014). Some advantages of HBase are low latency access to data, MapReduce and Hive/Pig integration, auto failover and reliability, and variable schema where columns can be added and removed dynamically (Teller, 2015). Disadvantages of HBase include no built-in authentication or permissions, is a single point of failure when only one MasterServer is used, and provides no transaction support (Shripurv, 2014).

HBase is a column-oriented database where HBase tables are stored in ColumnFamilies and each ColumnFamily can have multiple columns (Teller, 2015). Figure 13 represents the layout of HBase on top of Hadoop. The main components of HBase are the MasterServer, RegionServer, Region, and Zookeeper. The MasterServer is the administrator and is responsible for cluster monitoring and management, assigning Regions to RegionServers, and failover and load balancing by re-assigning the Regions (Teller, 2015). RegionServers are identified by the MasterServer and reside on DataNodes; they manage regions in coordination with the MasterServer, conduct data splitting in the Regions, and coordinate and serve the read/write process (Teller, 2015). Regions are used to manage the availability and data distributions and performs high velocity reads and writes (Teller, 2015). Zookeeper is used to monitor the RegionServers and recover them if they are down (Teller, 2015).

Figure 13. HBase Layout



Source: Teller, S. (2015). *Hadoop essentials*. Birmingham, UK: Packt Publishing, p. 260.

b. Hadoop Distributed File System

The amount of data collected by organizations has reach the level beyond what a single machine can handle, creating the need for a new method of storing data (White, 2015). File systems that store data across a network of machines vice maintaining it on a single local machine are called distributed file systems (White, 2015). The file system that comes with Hadoop is called the Hadoop Distributed File System (HDFS). HDFS is

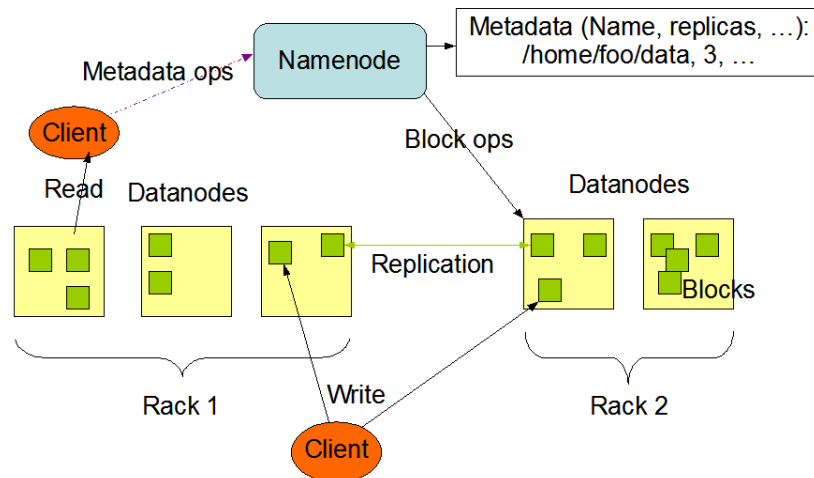
designed to operate on commonly available hardware and reliably store vast amount information (Borthakur, 2008; White, 2015). Commodity hardware means low cost, commonly available hardware that can be obtained from multiple vendors (White, 2015). When running a large cluster set, having commodity hardware is necessary because the chance of having a node failure is high. If a node does fail, HDFS will continue to operate as normal without noticeable interruption to the user (White, 2015). Very large data sets mean files that are hundreds of Megabytes, Gigabytes, or Terabytes in size (White, 2015). Lastly, HDFS utilizes the most efficient streaming data access pattern: write-once, read-many-times (White, 2015).

Though HDFS has many benefits when dealing with Big Data, it does have its shortfalls. HDFS cannot support low-latency access to data or random modifications to files by multiple users (White, 2015). HDFS is optimized for delivering a high throughput of data; therefore, applications requiring low-latency access to data will not work well with HDFS (White, 2015). The NameNode in Hadoop holds the file system metadata in memory, which limits the number of files HDFS can store (White, 2015). According to White (2015), storing millions of small files may be feasible but storing billions can see a degradation in performance due to the limitations of current hardware. Furthermore, the current configuration of HDFS does not support multiple writers.

(1) HDFS Architecture

HDFS operates in a master/slave architecture (Borthakur, 2008). Figure 14 shows a simple HDFS architecture. The master server of HDFS is called the NameNode, and it manages the name space of the file system and controls client access to files (Borthakur, 2008). According to Shvachko, Kuang, Radia, and Chansler (2010), the NameNode selects which DataNode to store blocks to and provides the client the locations of those blocks. Since NameNodes hold the locations of data blocks, they are critical to the operation of HDFS because without it, files cannot be reconstructed (White, 2015).

Figure 14. HDFS Architecture



Source: Borthakur, D. (2008). HDFS architecture guide. Retrieved Nov 25, 2015, from https://hadoop.apache.org/docs/r1.2.1/hdfs_design.pdf

DataNodes are the slaves in the HDFS architecture. The purpose of the DataNode is “to store and retrieve blocks when told to by a client or the NameNode” (White, 2015, p. 138). Additionally, DataNodes perform block creation, deletion, and replication (Borthakur, 2008). Periodically, DataNodes send reports called block reports notifying the NameNode what blocks it has in its possession, and heartbeats to signal that it is still operational and blocks are available (Shvachko et al., 2010). NameNodes will tag a DataNode inoperable if no heartbeats are received within ten minutes (Shvachko et al., 2010). Heartbeats are important because they allow the NameNode to efficiently balance the work load (Shvachko et al., 2010).

When requesting to read a file, a HDFS client first communicates with the NameNode requesting for a list of DataNodes that contains the blocks of the file (Shvachko et al., 2010). The client then contacts the respective DataNodes directly and requests the transfer of the desired block (Shvachko et al., 2010). When writing a file, the client again first communicates with the NameNode who decides which DataNode the client will write to (Shvachko et al., 2010). Then the client creates a connection to the DataNode and sends the data (Shvachko et al., 2010). Once the first block is filled, the client sends another request to the NameNode for another DataNode to write the second block and the process continues again until all data is written (Shvachko et al., 2010).

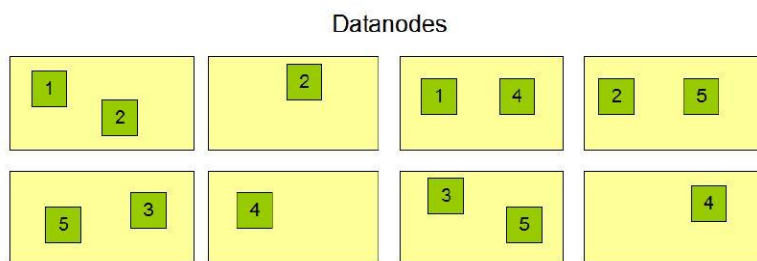
Unlike traditional file systems, HDFS provides an application program interface (API) that shows the locations of the file blocks, allowing applications to locate and retrieve data faster, improving the read performance of the system (Shvachko et al., 2010).

(2) Data Replication

When a file gets ingested into HDFS, they are broken up and stored as blocks with a default size of 128 MB (White, 2015). According to White (2015), HDFS differs from conventional file systems in that a file does not take up the entire block if it the size is less than the default size. For example, a 5 MB file stored on a 128 MB block uses only 5 MB of disk space, not 128 MB. Figure 15 shows that blocks are replicated among the DataNodes for fault tolerance (Borthakur, 2008). The number of times a data is replicated is set by application when the file is created or at future time (Borthakur, 2008).

Replica placement is a critical component to HDFS because it improves the availability and reliability of data stored on HDFS, and increases the performance by effectively utilizing network bandwidth (Borthakur, 2008). The default replication factor for HDFS is three; this placement policy places one replica in one local node and the other two on the same remote rack but in two different nodes (Borthakur, 2008). According to Borthakur (2008), this policy improves the write operation without any negative impact on the read process or the reliability of the data.

Figure 15. DataNode Replication



Source: Borthakur, D. (2008). HDFS architecture guide. Retrieved Nov 25, 2015, from https://hadoop.apache.org/docs/r1.2.1/hdfs_design.pdf

During replica selection, HDFS selects a replica closest to the reader to improve the bandwidth efficiency and to reduce the read time (Borthakur, 2008). In a situation

where the replica and reader are on the same node, HDFS will select that replica vice reaching out to another node (Borthakur, 2008). Replicas residing on local data centers are preferred when the HDFS is spread across many data centers (Borthakur, 2008).

(3) Common HDFS Failures

The main idea behind HDFS is to reliably store data regardless of system failures (Borthakur, 2008). NameNode, DataNode, and network partition failures are common failures found in HDFS (Borthakur, 2008). Since the hardware used in an HDFS cluster is commodity and low-cost, the probability of hardware failure is high. Failure of a network partition can potentially result in loss of connectivity between DataNode and NameNode (Borthakur, 2008). According to Borthakur (2008), a NameNode detects a loss of connectivity if it fails to receive a heartbeat message from a DataNode. Furthermore, a DataNode failure would automatically result in the NameNode to re-replicate lost blocks.

Data integrity is another type of failure that can be experienced in HDFS. Storage device failures, network issues, and bugs associated with poor software designs can result in DataNodes receiving corrupted data (Borthakur, 2008). According to Borthakur (2008), data integrity is maintained by computing checksums on each block of a file. After the checksum is computed, it is stored in a hidden file in HDFS (Borthakur, 2008). During file retrieval, the client computes another checksum on the retrieved file and compares it to the associated checksum in the hidden file (Borthakur, 2008). If the checksum does not match, the client decides whether to take in that file or retrieve another replica from a different DataNode (Borthakur, 2008).

4. Hadoop Infrastructure Layer

The purpose of the infrastructure layer is to support the storage layer (Sawant & Shah, 2013). Traditional data architectures provide strong transactional capability but trades away the ability to scale and are expensive (Sawant & Shah, 2013). On the other hand, Hadoop was designed to run on commodity hardware and still provide the strong transactional capability of traditional data architectures. By using commodity hardware, organizations are not limited to a single vendor (White, 2015). Instead, they are able to use standardized, commonly available hardware available from any well-known vendor

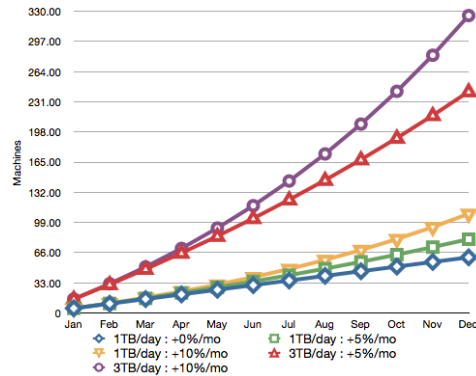
(White, 2015). Hadoop hardware come in two classes: masters and workers. Master nodes run the NameNodes, secondary NameNodes, and Jobtracker (Sammer, 2012). Though Hadoop is about using commodity hardware, selecting master node hardware is where organizations should plan to spend more money because it runs all the critical services for the cluster. For small clusters of fewer than 20 worker nodes, the recommended hardware profile for master nodes are a dual quad-core 2.6 GHz CPU, 24 Gigabytes of DDR3 RAM, dual 1 Gigabit Ethernet network interface cards (NIC), a SAS drive controller, and at least two SATA II drives in a just a bunch of disks (JBOD) configuration in addition to the host operating system device (Sammer, 2012). Mid-size clusters of up to 300 nodes should have at least an additional 24 Gigabytes of RAM for a total of 48 GB, and any clusters greater than 300 nodes should have a total of 96 Gigabytes of RAM (Sammer, 2012).

Some considerations to take when selecting worker node hardware is that they are responsible for both storage and computation; hardware must have enough storage capacity as well as enough CPU and memory to process data. For deciding on how much storage is needed, consider an organization that processes one Terabytes a day. By default, Hadoop will replicate the data three times for a total of three Terabytes a day. According to Sammer (2012), an estimate of 20–30 percent of a host's hard disk storage is reserved for temporary data. Therefore, if there were 12 machines with two Terabytes each, it leaves approximately 17 Terabytes of space to store HDFS data, or six days' worth of data. Selecting memory size is a little difficult as the reserve memory for Hadoop and the operating system plus the number of tasks that is completed need to be taken into account. Sammer (2012) states that a typical task uses two to four Gigabytes of memory. A machine with 64 Gigabytes of memory can support around 16 to 32 tasks. The recommended hardware profile for high end worker nodes are two, six core 2.9 GHz CPUs, 96 Gigabytes of DDR3 RAM, two, six Gigabytes/second SAS disk controller, 24 hard drives with one Terabyte of storage, and a ten Gigabit Ethernet NIC.

Once hardware configuration has been selected, the next step is to determine how many machines are required for each cluster. The most common method is basing the cluster size on the amount of storage needed (Sammer, 2012). If business processes

require a high data ingest rate, then more machines are needed. As more machines are added to a cluster, more computing power is added as well. Figure 16 shows a projection on how many machines with a disk size of 18 Terabytes will be needed with either a five or ten percent increase in data per month.

Figure 16. Cluster Size Growth Projection

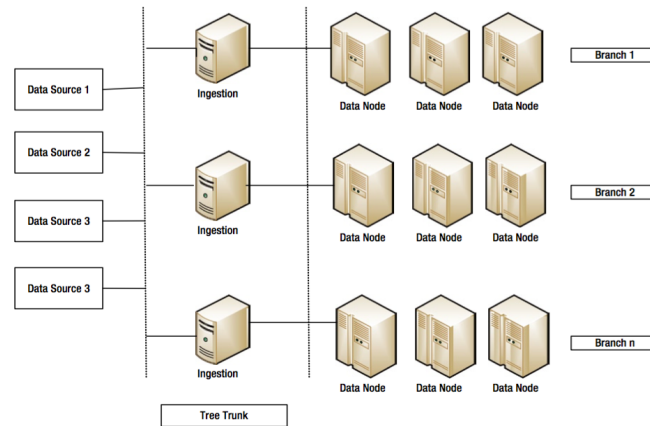


Source: Sammer, E. (2012). *Hadoop operations*. Sebastopol, CA: O'Reilly Media, Inc., p. 51.

The Hadoop Infrastructure Layer can either be physical, meaning located on-site of an organization, or virtual, using third-party cloud vendors. Figure 17 shows what a typical, on-site infrastructure would look like. It is an N-tiered tree network and is the predominant architecture deployed today in data centers (Sammer, 2012). The number of tiers required is dependent upon the number of hosts that is required to support; two-tiered network can support 576 hosts while a three-tiered network can support 1,152 hosts (Sammer, 2012). Instead of building a Hadoop cluster on-site, it can be built virtually through cloud computing. One significant implication of a virtual Hadoop cluster is storage. A core architectural design of HDFS is that its replication process is low cost and a very effective way of storing large amounts of data (“Virtual Hadoop,” 2013). This design was based on a physical topology where Hadoop smartly places data across the cluster as a fail-safe for any host or rack failure. Some cloud vendors do not show the underlying physical topology to their customers making the original design of HDFS invalid (“Virtual Hadoop,” 2013). Data loss is more likely and sometimes unrecoverable, as the user will not know the exact location of data. Benefits to virtualizing Hadoop are

clusters can be set up, expanded, or contracted on demand, one image is cloned reducing operating costs, physical infrastructure can be reused, and costs are related to CPU usage (“Virtual Hadoop,” 2013).

Figure 17. Typical Big Data Hardware Topology



Source: Sawant, N., & Shah, H. (2013). *Big data application architecture Q&A: A problem - solution approach* (1st ed.). Berkeley: Apress, p. 16.

5. Hadoop Platform Management Layer Analytics

Technology has pushed the boundaries and limits of Information Science (IS) and provided enterprises a new view into the information world. Everyday organizations are continuing collecting information at an increasing rate. Enterprises are in need of tools that are capable of the manipulating vast amount of information as fast as possible. Luckily, the information industry has answered the call with many different business intelligence tools. This section will provide a brief description of the models and tools used by the management layer to manipulate and process large quantity of the information. These models and tools include MapReduce, Pig, Hive, Impala, Mahout, and Zookeeper.

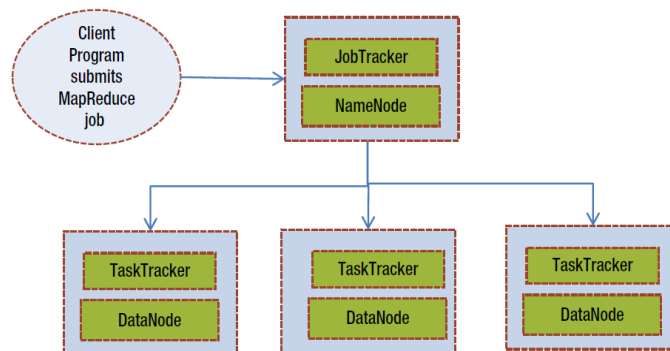
a. MapReduce

Since Hadoop deals with various types of data, it requires a robust application that will process such data. MapReduce was created to address that issue. It is a programming model designed to process Terabytes of data in parallel on Hadoop clusters (Apache,

2013). MapReduce uses batch mode to efficiently process structured and unstructured data (Sawant & Shah, 2013). The programming languages that MapReduce uses are Java, Ruby, Python, and C++ (White, 2015). The MapReduce process consists of two tasks: Mapping and Reducing. Mapping deals with taking the original set of data and breaking it down to a smaller set that is then used by the Reducing process (IBM, n.d.). Reducing then collects the smaller data set and combines them into a single file and stores it into HDFS (IBM, n.d.).

MapReduce takes advantage of Hadoop's architecture by distributing its workload across many nodes and clusters. This allows for an effective way to balance the workload and manage recovery from system failures (Sawant & Shah, 2013). The MapReduce task is consist of one master node called the jobtracker and a slave node called the tasktracker located one per node (Apache, 2013). According to Sawant and Shah (2013), the master node is responsible for assigning and tracking jobs to task tracker nodes as seen in Figure 18. Additionally, the master node is responsible to balance the work across many nodes, and reassign jobs in case of failures. (Sawant & Shah, 2013).

Figure 18. MapReduce Task



Source: Sawant, N., & Shah, H. (2013). *Big data application architecture Q&A: A problem - solution approach* (1st ed.). Berkeley: Apress, p. 18.

b. Pig

According to White (2015), there are two parts to Pig: a scripting language called Pig Latin and the execution environment. Pig Latin allows programmers to process large

data sets by manipulating data in HDFS (Sawant & Shah, 2013). Additionally, it is designed to run as a client-side application that is capable of interacting with HDFS to launch programs and process data (White, 2015). Applications like MapReduce rely on Java as the programming language to execute applications, which is very time-consuming and challenging. However, Pig Latin makes it easier to run programs with less lines of code (Sawant & Shah, 2013). Pig Latin also provides the ability to create richer data structures allowing for a more powerful way of manipulating data sets (White, 2015). On the other hand, the execution environment runs Pig Latin programs, which provides programmers the ability to write queries, and process Terabytes of data (White, 2015). Where Pig falls short is in the processing and handling small amount of data (White, 2015).

According to White (2015), Pig is designed to run in either in a local host file system or in HDFS with multiple clusters. The local mode allows Pig to run within local file system and is mainly used for testing scripts written for Pig. On the other hand, the MapReduce mode allows the execution of Pig's scripts and queries within HDFS and translate them into MapReduce jobs (White, 2015). The local and MapReduce mode can execute Pig scripts utilizing three settings: Script, Grunt, and embedded (White, 2015). Programmers are able to take advantage of the script mode to run script files, run grunt mode to use interactive shell to run Pig commands, or take advantage of the embedded mode to run Java codes using PigServer (White, 2015).

c. Hive

Hadoop architecture is designed to handle large volumes of data, and one way to help the architecture to perform this important task is the use of Apache Hive. Hive is a data-warehouse tool that allows the use of SQL environment within Hadoop ecosystem, and the ability to query stored data in HDFS using SQL (Sawant & Shah, 2013). The data-warehouse framework is installed on top of Hadoop architecture. Apache Hive uses a SQL like language called HiveQL, which makes managing and querying data a lot easier (Apache, 2014a). Additionally, Apache Hive provides Hadoop greater compression of the stored data resulting in more efficient storage utilization without any negative

impact on access speed (Sawant & Shah, 2013). Although Hive is designed to allow Hadoop to process and analyze large amount of data, it has its shortfalls. It is not a good tool to use for creating advanced machine-learning systems (White, 2015). On the other hand, an advantage of using Hive is the ease of installation. Hive is designed to run on workstations and provide programmers a way to convert SQL queries into MapReduce jobs that can be executed on Hadoop nodes and organize data into tables in HDFS (White, 2015).

d. Impala

Apache Impala is an open source analytic query engine that utilized distributed massively parallel processing (MPP) to run SQL queries on HDFS and Cloudera Distribution of Hadoop (CDH) (Cloudera, 2016). According to Henschen (2013), analyzing data using MapReduce is a very slow process. Impala has the capability to integrate within Hadoop's ecosystem and execute queries much faster than the MapReduce and Hive (Russell, 2014). Hive excels at processing batch queries that require long time to process; however, Impala is the excellent tool to use for running queries that require immediate results (Wadkar & Siddalingaiah, 2014). Impala is designed to work with the business intelligence tools that are based on a SQL model and quickly provide important results vital to the organizations (Russell, 2014). Impala uses simple SQL syntax to process complicated queries, and produce results in minutes (Russell, 2014). Another advantage of using Impala is the ability to work with all sorts of file formats, which allows the ability to work with other Hadoop tools without the need to convert file formats (Russell, 2014).

Impala is a great tool to use for this research since most of the data received is in text format and SQL queries are the perfect tools to analyze text information. Additionally, since the Navy MPTE data is already imported into the HDFS, the Impala can directly query the raw data without the need to load and organize for processing.

e. Mahout

Apache Mahout is the library of machine learning algorithm that utilizes clustering, collaborative filtering, and classification to process large data within HDFS

(Tiwary, 2015). Processing stream of data generated by online applications on a daily basis as quickly as possible can provide enterprises the edge they need to be successful. Mahout is the framework that learns from the existing data and makes predictions, without being explicitly programmed, which significantly improve the performance (Withanawasam, 2015). Mahout operates within the Hadoop ecosystem that provides for the scalability, and works alongside the other tools such as MapReduce (Withanawasam, 2015). Mahout uses a user-base and item base approaches to handle processing data from different streaming data sources without the need to convert format (Cloudera, 2013).

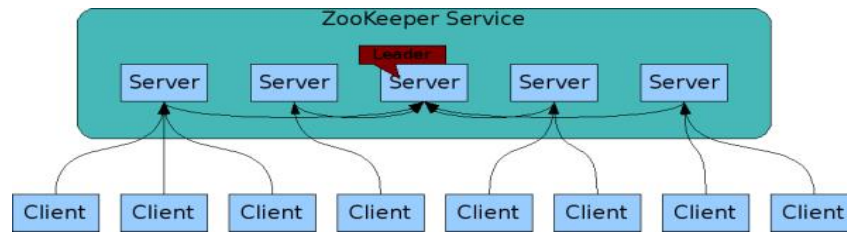
Mahout's capability to learn from existing data to make prediction on future outcomes makes it a useful tool to use, since the goal of this research is the ability to analyze the existing information from Aviation community, and use it to make prediction for future retention.

f. Zookeeper

According to White (2015), Zookeeper is an open source library that is capable of processing simple operations. A challenge that the Hadoop infrastructure most often run into is the synchronization between multiple nodes. According to Sawant and Shah (2013), the Zookeeper topology was designed to keep various Hadoop instances and nodes in sync with each other. According to Sawant and Shah (2013), the topology, as seen in Figure 19, provides protection from nodes failures in the Hadoop architecture. The distribution system provided by the Hadoop relies on coordination and handling of the partial failures between nodes, which requires topologies such as Zookeeper to handle the failures (Sawant & Shah, 2013).

The fault protection characteristic of Zookeeper makes it an important part of the Hadoop architecture. According to Sawant and Shah (2013), the Zookeeper topology takes advantage of a variety of tools to effectively and safely handle failures and reassigning the leader or primary node because they are interconnected. Zookeeper guarantees the Hadoop architecture data consistency by taking advantage of the qualities such as sequential consistency, atomicity, durability, single system image, and timeliness (White, 2015).

Figure 19. Zookeeper Topology



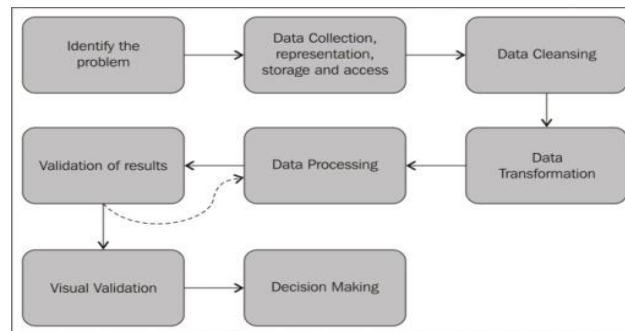
Source: Apache. (2014b). Zookeeper [Information on page]. Retrieved from <http://zookeeper.apache.org/doc/trunk/zookeeperOver.html>

6. Analytics Engine

Information has become an essential part of any enterprises operating in information world. Many of the large enterprises currently involve in the information field has recognized the importance of the data and the role it plays in success of their business. Their successes greatly rely on how fast they are capable of driving decision based on large amount of information available to their analyst. Unfortunately, large amount of information can lead to confusion, so it is very imperative to have the capability to quickly analyze information in various visual modes (Sawant & Shah, 2013). Hadoop architecture is the answer to information overload problem.

Hadoop takes advantage of variety of analytic tools to make sense of the collected information. The collected information is either data at rest, or streaming data. These tools follow the data analytics workflow as presented in Figure 20.

Figure 20. Data Analytic Workflow



Source: Karanth, S. (2014). *Mastering Hadoop*. Birmingham, UK: Packt Publishing, p. 298.

According to Karanth (2014), the first step in the process is to identify the problem following by the collection of the information that is related to the established problem. For this study the retention in the aviation community is the identified issue, and the collected information is from Navy MPTE databases. The next step is the cleansing the data to make sure any missing values that could potentially create skewed results are removed, and the data is transformed for processing (Karanth, 2014). Once the data is processed and validated, the results are prepped for visual presentation to the decision makers (Karanth, 2014).

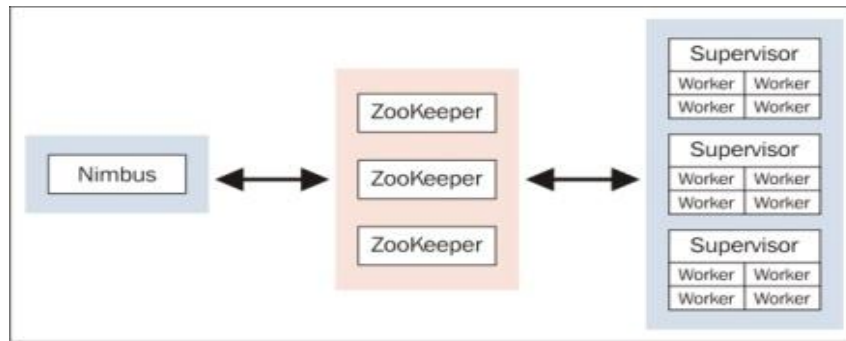
a. Data at Rest

Data at rest is referred to the information that is stored on HDFS. The Hadoop architecture utilizes tools such as MapReduce, Mahout to analyze the information. Apache Impala is one other tool used within the Hadoop ecosystem that utilizes SQL queries to provide text analysis (“Apache Impala,” n.d.). due to the vast amount of information stored within HDFS, Hadoop requires fast and capable search engines for “iterative and cognitive data discovery” (Sawant & Shah, 2013, p. 22). Open source search engines such as Apache Lucene and Solr are widely used by the industry because of their capability to quickly process data (Apache, 2016a).

b. Streaming Data

Streaming data is referred to the information that is generated consistently by machines, and need to be processed immediately to maintain their quality and integrity (Teller, 2015). Traditional data warehousing is mainly designed to allow batch processing information, which is not a useful model for processing the online data in real time (Oliver, 2014). Processing online information requires a very fast search engine. Apache Storm is a quick, real time search engine capable of real time analytics, online machine learning, and continuous computation (Apache, n.d.-b). Storm consists of a Master node (Nimbus) that utilizes the Zookeeper topology to distribute jobs across multiple supervisor nodes (Teller, 2015). Figure 21 depicts the physical architecture of Storm. Each supervisor node controls multiple worker nodes, and assign tasks received from Nimbus (Teller, 2015).

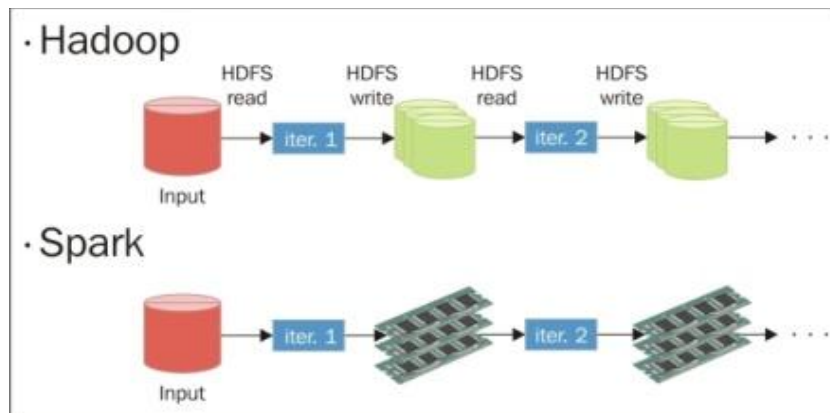
Figure 21. Physical Architecture of Apache Storm



Source: Teller, S. (2015). *Hadoop essentials*. Birmingham, UK: Packt Publishing, p. 406.

Apache Spark is another fast search engine capable of running programs up to 100x faster than Hadoop MapReduce (Apache, n.d.-a). Spark utilizes the memory to store and process information as presented in Figure 22.

Figure 22. Spark versus MapReduce



Source: Teller, S. (2015). *Hadoop essentials*. Birmingham, UK: Packt Publishing, p. 419.

7. Visualization Layer

Visualization layer is design to handle multiple roles within Hadoop ecosystem, which consists of Hadoop administration and visualization functions. Administration functions deal with processing tasks within Hadoop clusters, while visualization tools handle the output and presentation of the data. Large amount of information can lead to confusion, so it is very imperative for data scientist to be able to analyze information in various visual modes as fast as possible (Sawant & Shah, 2013). Hadoop's open source

design allows third party applications to provide organizations with a wide variety of visualization tools. Hadoop User Experience, Qlikview, Karmasphere, Datameer, and Tableau are some of the most popular visualization tools currently used within the Hadoop ecosystem. Visualization tools allows analyst make sense of the information ingested into Hadoop ecosystem by working inside Hadoop architecture and accessing the processed information. The visualization layer provides the compatibility to support traditional BI and Big Data in single consolidated view (Sawant & Shah, 2013).

Traditional analytic tools and techniques lack the capability to provide an overall picture for large amounts of data (Sawant & Shah, 2013). To overcome this challenge, new analytic techniques were designed to provide data scientists the ability to better understand processed information. Some examples of the new techniques are mashup view pattern, compression pattern, zoning pattern, first glimpse pattern, exploder pattern, and portal pattern (Sawant & Shah, 2013).

Analyzing large quantities of information using MapReduce can be time consuming; therefore, a technique such as the mashup view pattern was created to take advantage of the Hive layer and treat it like a virtual data center to combine, store, and quickly analyze information (Sawant & Shah, 2013). Instead of combining the data into one large virtual data center, analysts can use a compression pattern technique to transform information into a form that is easy and fast to access and process (Sawant & Shah, 2013). Another technique used to decrease access time to data is the zoning pattern, which breaks and indexes data into relative smaller blocks based on different characteristics (Sawant & Shah, 2013). Data analyst use First Glimpse technique to only view small but necessary information in the interim state, and have the ability to access full information if needed (Sawant & Shah, 2013). Exploder pattern works in the same manner as the first glimpse but allows the different view of data from separate data source (Sawant & Shah, 2013). Portal pattern allows the use of existing enterprise tool by improving the visualization feature to provide better representation of the big data (Sawant & Shah, 2013).

8. Security and Monitoring Layer

Information is a very powerful element of success for any enterprises. The more information available to the organization could potentially result in better decisions involving the enterprise's future and success. The stored information varies in content, and could possibly contain sensitive data regarding the customers. According to Teller (2015), the information is extremely valuable to companies and must be protected to the best of their ability, especially if the information contains sensitive data.

Information security focuses on the protection of the data stored within organizational databases. The majority of the cybercrimes committed in recent years involve data theft by either outside threats or from within organizations (Teller, 2015). Outside threats are referred to the hackers from outside of the organization attempting to target information stored on Hadoop clusters to gain access to valuable information (Sawant & Shah, 2013). Studies conducted by the professionals in information security have indicated the majority of the data thefts are inside security threats from within organizations, and committed by either disgruntle or untrained employees (Teller, 2015).

Hadoop security infrastructure has been developed over the years, and is currently based on the enterprise security model that evolved around four security pillars (Shukla, 2013). These implemented security pillars are authentication, authorization, auditing (Accounting), and data protection. The authentication in Hadoop is done through two methods. The first method is the simple authentication or pseudo-authentication, which relies on user's assertion to prove their identity (Teller, 2015). The second method takes advantage of the Kerberos system, and provides a fully secure Hadoop cluster as a single point of authentication (Teller, 2015). Authorization is the process of granting authenticated users the appropriate privilege to access the files. According to Shukla (2013), authorization within Hadoop clusters is managed by the HDFS at the resource level, and by the MapReduce at the service level. Auditing is referred to the ability of the system to keep track of the access and manipulation of the data stored within an organization's data storage. In case of data breaches, auditing provides valuable forensic information to assess the damage, which could lead to identifying the guilty party (Teller, 2015). Confidentiality and privacy are two of the most important characteristic of the

information. The Hadoop architecture utilizes the cryptographic methods to protect data in transit across the network, and OS-level encryption to protect information data at rest within HDFS (Teller, 2015).

9. Hadoop Distributions

The Apache Hadoop distribution is the open source version of Hadoop. Support for Apache Hadoop is done through online forums where questions are addressed to the community and answered by its members. Deployment and management of Apache Hadoop distribution is not easy. Additionally, Hadoop is written in Java and optimized to run on Linux systems, which may not be compatible with existing enterprise applications and infrastructures. To address the difficulties of implementing Apache Hadoop, companies have started developing their own variations of Hadoop. Distributions come in three types. The first type is providing commercial or paid support and training for Apache Hadoop distribution (Karanth, 2014). Second is companies providing a set of supporting tools for deployment and management of Apache Hadoop (Karanth, 2014). Third, companies supplement Apache Hadoop with proprietary features and code (Karanth, 2014).

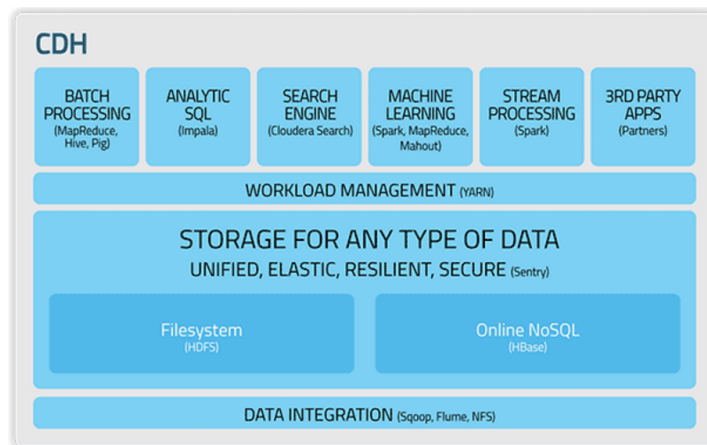
When selecting a distribution, organizations should take into account the following attributes: performance, scalability, reliability, and manageability. Performance is defined by a cluster having high throughput, and the ability to ingest input data and emit output data at a quick rate (Karanth, 2014). Over time, data will outgrow the physical capacity of an organization. Scaling out should be as easy as adding more nodes, but distributions could place a burden on it. Scaling costs would depend upon the existing architecture and how it compliments and complies with the Hadoop distribution (Karanth, 2014). Distributed file systems are subject to failures. Distributions that eliminate single points of failures and reduce manual tasks for cluster administrators tend to be more reliable (Karanth, 2014). What differentiates most distributions are the capabilities of Hadoop management tools. These tools need to provide centralized administration, resource management, configuration management, and user management (Karanth, 2014). There are a number of distributions of Hadoop available. For this research, we

examine Cloudera Distribution of Hadoop, IBM BigInsights, and Amazon Elastic MapReduce.

a. *Cloudera Distribution of Hadoop*

The Cloudera Distribution of Hadoop (CDH) is among the most popular utilized distributions because of its customer support and applications such as Cloudera Manager (Teller, 2015). Cloudera Manager is a web-based user interface that creates, manages, and maintains a Hadoop cluster. CDH includes many core applications from the Hadoop ecosystem such as HBase, MapReduce, Pig, Hive, and Zookeeper as seen in Figure 23. The Navy currently adopts CDH for the Naval Tactical Cloud.

Figure 23. Cloudera Distribution of Hadoop



Source: CDH overview. (2016). Retrieved from http://www.cloudera.com/documentation/enterprise/latest/topics/cdh_intro.html

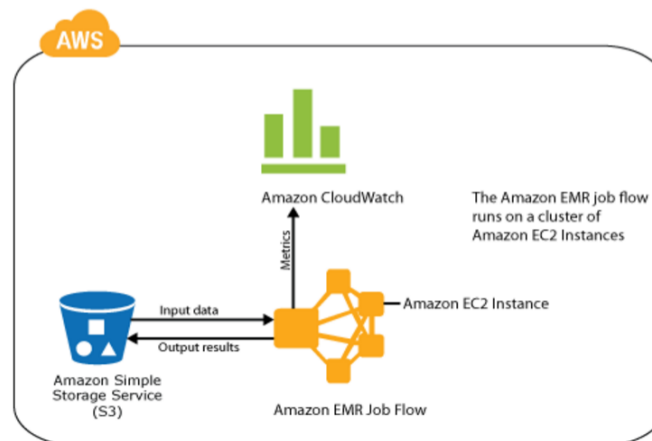
b. *IBM InfoSphere BigInsights*

IBM InfoSphere BigInsights is designed to bring Hadoop to the enterprise. It is a fast, robust, and easy-to-use platform for analytics on Big Data at rest (Zikopoulos, Eaton, deRoos, Deutsch, & Lapis, 2012). IBM designed BigInsights to make management easy through their graphical installation, configuration, and administrative tools (Zikopoulos et al., 2012). Additional, it provides an industry-leading text analytics toolkit and engine (Zikopoulos et al., 2012).

c. *Amazon Elastic MapReduce*

Through Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3), Amazon leverages their comprehensive cloud services to provide Amazon Elastic MapReduce (EMR). Figure 24 illustrates how Amazon EMR operates with different Amazon EC2 services. Amazon EMR provides organizations the ability to store Petabytes of information by distributing workload thru virtual servers within Amazon cloud environment (Amazon, 2015). As with other distributions, Amazon EMR provides several analytical applications from the Hadoop ecosystem. Benefits of running Amazon EMR are the ability to scale up or down when necessary and manage virtual servers instantaneously (Amazon, 2015).

Figure 24. Amazon EMR Interaction with Other Cloud Services



Source: Amazon. Amazon Elastic MapReduce: Developer guide. (2015). Retrieved from <http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-what-is-emr.html>

C. CHAPTER SUMMARY

This chapter described in detail different data warehouses and components of a Big Data Application architecture. The core components of Hadoop are HDFS, NoSQL database, Hbase, and several applications used for analytics. We then examined three commonly used Hadoop distributions. The next chapter will focus on the process of conducting a predictive analytics project.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. CROSS-INDUSTRY STANDARD PROCESS FOR DATA MINING

Predictive analytics projects require a framework to guide data analysts in developing an effective model. The Cross-Industry Standard Process for Data Mining (CRISP-DM) will be used as the framework to build a predictive model to answer Navy MPTE's retention questions for the post-command aviator community. According to Abbott (2014), data analysts have favored CRISP-DM over any other process model since it was first created in the 1990s. CRISP-DM's popularity is due to the fact that it is a good representation of how data mining projects are conducted in the real world (Chapman et al., 2000). The purpose of this chapter is to gain an understanding of the CRISP-DM process prior to applying it on our data set on post-command aviators.

A. CRISP-DM OVERVIEW

CRISP-DM was created in 1996 by experts from three up-and-coming companies in the data-mining market: DaimlerChrysler, SPSS, and NCR (Nava & Hernández, 2012). The early 1990s showed a growing need for data mining, and every data mining user was creating his or her own methods to data mining, which sparked the need for a standard process model that is freely available (Nava & Hernández, 2012). After obtaining funding from the European Commission, a CRISP-DM Special Interest Group was launched to get input from data-mining practitioners from around the world and other interested parties like data warehouse vendors (Nava & Hernández, 2012). Two years after the Special Interest Group convened, multiple trials of CRISP-DM were ran on large-scale data mining projects for companies like Mercedes-Benz (Nava & Hernández, 2012). By the end of the decade, CRISP-DM 1.0 was released and tested with huge success by DaimlerChrysler, SPSS, and NCR (Nava & Hernández, 2012).

The audience for CRISP-DM is not only data analysts but also program managers. CRISP-DM provides programs managers into the predictive modeling process by revealing the steps their data analyst will conduct (Abbott, 2014). During each step of the process, the program manager can track cost estimates to ensure deliverables and

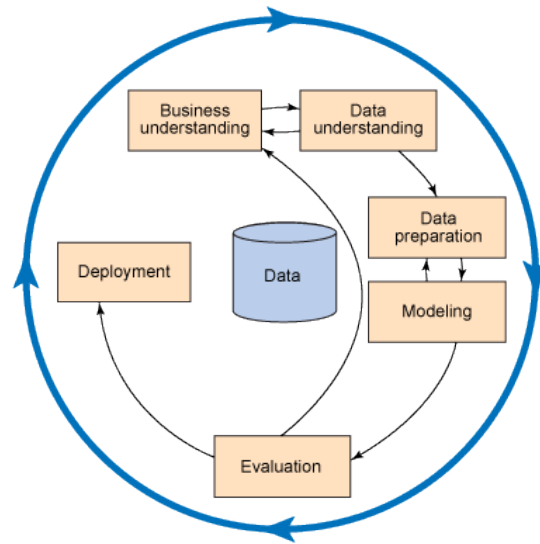
timetables are meeting deadlines (Abbott, 2014). In addition, for many sub-tasks, a report is created describing what decisions were made and the reasoning for those decisions, assisting program managers in further understanding the process (Abbott, 2014).

For data analysts, CRISP-DM details each step of process providing a structure for analysis, reminding the analyst what steps need to be taken and what needs to be documented or reported (Abbott, 2014). Though CRISP-DM describes the modeling process in a linear fashion, this is nearly never the case as problems always arise in projects that necessitate applying the process iteratively. CRISP-DM provides gives data analysts a good baseline to refer to when reporting to program managers (Abbott, 2014).

B. CRISP-DM PHASES

There are six phases to the CRISP-DM process as shown in Figure 25: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. The arrows in the diagram denotes areas where the process is modified based on the findings during the project. For example, if it was found that during the Data Understanding step that the data quantity or quality is insufficient, then it would be necessary to return back to the Business Understanding step to redefine business objectives based on the available data. The following sections will describe the specific tasks involved in each phase of CRISP-DM.

Figure 25. CRISP-DM Process Model



Source: Chandran, S. S. (2014). *Making better business decisions with analytics and business rules: An overview of the IBM decision management product stack*. Retrieved from http://www.ibm.com/developerworks/bpm/library/techarticles/1407_chandran/1407_chandran-pdf.pdf

1. Business Understanding

Any predictive modeling project requires clearly identified objectives. The Business Understanding phase assists organizations in determining these objectives in business terms. According to Chapman et al. (2000), there are four tasks in the Business Understanding phase: determining business objectives and data mining goals, assessing the situation, and producing a project plan.

a. Determine Business Objectives

Determining business objectives require data analysts to clearly comprehend what the organization wants to achieve from the project (Chapman et al., 2000). In addition, this first task will discover important factors that can possibly affect the outcome of the project (Chapman et al., 2000). Three entities of an organization must collaborate to determine business objectives: domain experts, data or database experts, and predictive modeling experts. According to Abbott (2014), domain experts have a good understanding of organizational processes and define the problem for analysis.

Furthermore, data or database experts identify the data needed to create a predictive model and how it will be accessed. In addition, predictive modelers develop the models that will achieve business objectives. If business objectives are not clearly defined, models that are built would go unused as it would not be useful to the organization. It is important that this step does not get overlooked to avoid repeated work.

b. Assess the Situation

The second task this phase is to assess the situation. According to Chapman et al. (2000), this involves identifying resources such as personnel, data, and any computing hardware or software that will be used in the project. Also identified in this task is any requirements such as scheduled completion dates, assumptions that may include whether or not the available data can achieve data mining goals, and constraints such as the limitations in available computing hardware (Chapman et al., 2000).

c. Determine Data Mining Goals

According to Chapman et al. (2000, p. 16), “data mining goals states project objectives in technical terms.” Data mining goals are determined by looking at available data and how it will be used to answer business objectives.

d. Produce Project Plan

The last task is to produce a project plan. According to Chapman et al., (2000), detailed steps will be laid out on how data mining goals and business objectives will be achieved. Furthermore, the project plan will also include a description of the initial set of tools and techniques that will be used.

2. Data Understanding

According to Chapman et al., (2000), the Data Understanding phase is where the initial data is collected, described, explored, and verified for quality. In addition, hypotheses are made to determine whether the initial data set will accomplish business objectives and data mining goals.

a. Collect Initial Data

The first task in this phase is to collect the data that will be used for analysis (Chapman et al., 2000). Data loading is also part of initial data collection and may assist in further understanding the data (Chapman et al., 2000).

b. Describe Data

According to Chapman et al., (2000), once the data has been collected, the next task is to examine it. This includes looking at the format of the data, the number of available records and fields in each table, the descriptions of each field, and any other features that is worth noting.

c. Explore Data

After careful examination, the data is explored and data mining questions are investigated through the use of querying, visualization, and reporting (Chapman et al., 2000). Exploration of the data also includes looking for any obvious patterns or relationships that may answer data mining goals or assist in the transformation of data or any other steps that are part of Data Preparation (Chapman et al., 2000).

d. Verify Data Quality

The last step in this phase is data verification and will look into the completeness and accuracy of the data, which includes identifying errors and missing values (Chapman et al., 2000). Identifying missing values are important as they can be represented in numerous ways such as null, zero, or spaces. Converting missing values into a single representation are done in the Data Preparation phase.

3. Data Preparation

According to Abbott (2014), about 60-90 percent of the project's time can be devoted to the Data Preparation phase. The tasks in this phase are selecting, cleaning, constructing, integrating, and formatting the data with the purpose of preparing the final data set for analysis (Chapman et al., 2000). These tasks can be repeated multiple times.

a. *Select and Clean Data*

The first task of this phase is to select the final data set that will be used for analysis (Chapman et al., 2000). According to Chapman et al. (2000), some considerations to take when selecting the final data set are how it relates to data mining goals and if there are any limitations on data volume or types. Next, the selected data set is cleansed. According to Abbott (2014, p. 21), data cleansing involves fixing problems in the data that was found in Data Understanding in order to improve data quality to a “level required by the selected analysis techniques.”

b. *Construct, Integrate, and Format Data*

Once the data is cleansed, it may be necessary to derive new attributes from existing attributes or create entirely new records (Chapman et al., 2000). A data set may contain multiple tables and it may be necessary to combine the tables to create a single table that will be run through the model (Chapman et al., 2000). Finally, data may need to be reformatted as acquired by the modeling tool; for example, some tools require that the first field of a table be a unique identifier or limiting all values to a certain number of characters (Chapman et al., 2000).

4. *Modeling*

The Modeling phase is where several modeling techniques are selected, a test design is generated, and the model is built and assessed (Chapman et al., 2000). According to Chapman et al., (2000), not all techniques use the same data formats; therefore, analysts may need to revisit the Data Preparation phase if several techniques are used.

a. *Select Modeling Technique*

Due to the wide variety that may be available for analysis, several models could be applied to the data set such as decision trees or regression analysis (Olsen & Delen, 2008; Chapman et al., 2000). Model selection should be based on which could meet the business objectives as defined in the Business Understanding phase (Chapman et al., 2000).

b. Generate Test Design

According to Chapman et al. (2000), when a modeling technique has been selected, it has to be tested to ensure its quality and validity. Testing involves separating the data sets into train and test sets. Train sets are used to build the model, while test sets are used to determine if the model is adequately designed (Chapman et al., 2000).

c. Build and Assess Model

Once the model or models have passed the test design, the prepared data set is applied onto the modeling tool (Chapman et al., 2000). The last step in the Modeling phase is to assess the model and rank them if multiple models were used (Chapman et al., 2000). Models are first judged technically then domain experts are brought in to interpret the models in business terms and determine if it will be valuable to the organization (Chapman et al., 2000).

5. Evaluation

In the Evaluation phase, final approval of a model is made based on whether it meets the business objectives as defined in the Business Understanding phase (Chapman et al., 2000). The tasks involved in the Evaluation phase are evaluating results, reviewing the process, and determining the next steps (Chapman et al., 2000).

a. Evaluate Results

According to Chapman et al. (2000), an important objective of this phase is to determine whether there is a reason why a model does not meet business requirements. In addition, this task may also reveal whether other data mining results are generated vice the ones already established. These newly revealed data mining results can cause additional challenges or unveil new information (Chapman et al., 2000).

b. Review Process

Chapman et al. (2000) stated that the purpose of the review process is to ensure that any important tasks or factors have not been overlooked. Furthermore, part of the

review process is quality assurance, which looks at whether the model was built correctly or if the right attributes were used

c. Determine Next Steps

Finally, the based on the evaluation and review process, the next steps of the project are determined whether it is to finish the project and continue onto deployment, refine the model, or start fresh with a new project (Chapman et al., 2000).

6. Deployment

If the model passes the Evaluation phase, the project then moves onto the Deployment phase where plans for deployment, monitoring, and maintenance are developed (Chapman et al., 2000). Other tasks in the Deployment phase include producing the final report and a final review of the project (Chapman et al., 2000).

a. Plan for Deployment, Monitoring, and Maintenance

When planning for deployment, results from the Evaluation phase are used to create a strategy for the deployment of the model in the organization (Chapman et al., 2000). In addition to the deployment strategy, a plan for monitoring and maintenance is also created, which is important if the model will be used in the day-to-day operations of the organization. According to Chapman et al. (2000), the purpose for monitoring and maintenance is to avoid a long-term misuse of data mining results.

b. Produce Final Report and Review Project

Upon completion of a project, a final report is generated which may contain a summary of the entire project or a presentation of the data mining results (Chapman et al., 2000). The last steps of the Deployment phase are to interview all key personnel involved in the project, focusing on their experiences and recommendations for future projects (Chapman et al., 2000). Personnel interviews are compiled into a final review of the project, which summarizes all lessons, learned (Chapman et al., 2000).

C. CHAPTER SUMMARY

This chapter examined each phase of the CRISP-DM process in preparation of applying it on the data set of post-command aviators. In the next chapter, we apply CRISP-DM process to the post-command aviator data set.

THIS PAGE INTENTIONALLY LEFT BLANK

V. APPLICATION OF CRISP-DM TO THE POST-COMMAND AVIATOR COMMUNITY

In this chapter, we applied the CRISP-DM methodology presented in the previous chapter to the post-command aviator community data. The objective was to determine whether a predictive model for aviator retention could be built using the data set available from NMPBS. First, we had to gain a business understanding of Navy MPTE in order to determine business objectives and data mining goals. Then we collected and examined the data set of post-command aviators extracted from NMPBS to identify quality issues, and form initial insights and assumptions for the development of a predictive model. Next, we cleaned, filtered, and transformed the data set by removing existing fields, adding new fields, and modifying data formats in preparation for final analysis. Once the data were ready for analysis, we applied several models and selected the best option that meets the business objectives and data mining goals. We then evaluated the results of the models to determine how accurately they achieved the business objectives.

A. BUSINESS UNDERSTANDING

1. Business Objectives

A detailed background of Navy MPTE's organization, processes, systems, and challenges was described in Chapter II. One of their major concerns was the retention of post-command aviators. Navy MPTE leadership found that aviators were leaving the service following their initial squadron command tour and pursuing careers in the commercial airline industry. The Navy invests substantial amount of resources in an aviator who screens for command. Their experience, education, and training make them valuable human capital that the Navy would like to retain for eventual service in a major command billet. One of the Navy MPTE's business objectives is to predict the retention of the post-command aviator community. By doing so, Navy MPTE leadership will be able to make policy changes in order to retain talented officers.

2. Assess Situation

Personnel databases available for this study included OPINS, NES, NTMPS, and NSIPS; however, we obtained data only from NMPBS, the web interface for OPINS. An advantage of NMPBS is that it contains historical data that would allow building models using supervised data mining techniques. To retrieve post-command aviator records from NMPBS, we filtered out officers who obtained the Navy Officer Billet Classification (NOBC) code of 8670, which signifies that they served a tour as a commanding officer of a squadron (OPNAV N13, 2015).

A constraint that affected the research was access to the OPINS database. Due to the sensitive nature of the data, we employed measures to safeguard the security of personally identifiable information (PII) security. Proper training, access forms, and requests for data were required in order to acquire the data necessary to perform the analyses.

3. Determine Data Mining Goals

The data mining goal of this study was to develop a model to predict retention of post-command aviators to support the MPTE business objective of increased retention. In addition, this research will provide a ranking of the importance of indicators that will help predict whether an aviator will remain in the service long enough to be screened for major command.

4. Produce Project Plan
















Given the limited size of the data set, the research used IBM SPSS Modeler 17.0 as the primary tool to perform predictive analyses using the CRISP-DM process. By applying supervised learning to historical data with known target field values, the modeler created a number of models that predicted whether a post-command aviator would continue his/her career to achieve higher milestones.

B. DATA UNDERSTANDING

1. Data Collection

The first task in Data Understanding is data collection. Prior to extracting data from OPINS, we had to identify the fields that we believed would be valuable to the research, which we achieved by examining the OPINS data dictionary. Since OPINS data contain all officer records, we filtered out records that contained only the NOBC of 8670. The OPINS data was then exported into a Microsoft Excel formatted file for easy manipulation. To create a sufficient sample size, data were extracted beginning in 2002. To minimize the number of extractions, we used only data from December of each year with the exception of the year 2016, which included only the month of January. We then discovered that there were multiple entries for each record, and thus removed duplicate entries, keeping the most recent record. The final data set retrieved from NMPBS included 15 separate Excel files representing every year of data, as shown in Figure 26. The identifying key for each service member was represented by the PERSON_MD5 field. Finally, we merged the Excel files into a consolidated file to load into SPSS Modeler.

Figure 26. Extracted NMPBS Files

Name	Type	Size
 2002	Microsoft Excel C...	1,142 KB
 2003	Microsoft Excel C...	1,146 KB
 2004	Microsoft Excel C...	1,136 KB
 2005	Microsoft Excel C...	1,154 KB
 2006	Microsoft Excel C...	1,165 KB
 2007	Microsoft Excel C...	1,165 KB
 2008	Microsoft Excel C...	1,155 KB
 2009	Microsoft Excel C...	1,172 KB
 2010	Microsoft Excel C...	1,156 KB
 2011	Microsoft Excel C...	1,186 KB
 2012	Microsoft Excel C...	1,162 KB
 2013	Microsoft Excel C...	1,109 KB
 2014	Microsoft Excel C...	1,136 KB
 2015	Microsoft Excel C...	1,168 KB
 2016	Microsoft Excel C...	1,169 KB

2. Data Description

After removal of all duplicate records, the final data set used for data preparation yielded 2,550 records for both active and retired Naval Aviators and Naval Flight Officers, with 82 fields representing the different attributes of a record. Table 1 shows the fields retrieved.

Table 1. Retrieved Fields from NMPBS

Field Name	Data Type	Description
MONTH_DESC	String	Month and year the record was updated
PERSON_MD5	String	Identifying key for each service member
DESIGNATOR_DD	String	Designator of service member
ACBD	Timestamp	Date of commissioning
ADSD	Timestamp	Date entered active duty
SRCE_CD_PGM	String	Commissioning source code
CURRENT_GRADE	String	Current rank
ELG_RETIRE_DT	Integer	Year eligible to retire
SEPN_TRNS_LOSS_DT	Timestamp	Separation date
SEPN_SPD_CD	String	Separation code
TOTAL_ACTIVE_SERVICE_DD	Integer	Active service in years
TOTAL_ACT_COMMISSION_SERV_DD	Integer	Active commissioned services
ETHNIC_GRP_CD	String	Ethnic code
RACE_CD	String	Race code
SEX_CD	String	Sex code
DEPN_NUM_HOUSEHOLD	Integer	Number of dependents
ASGN_NOBC_1-12	String	Navy Officer Billet Classification
ASGN_DPLY_DURA1-12	Integer	Time spent on deployment during tour
YEAR_GROUP_DD	Integer	Year commissioned
PROG_PROM_HIST_DT_I_LCDR	Timestamp	Date of rank for Lieutenant Commander
PROG_PROM_HIST_DT_H_CDR	Timestamp	Date of rank for Commander
MASTERS_ONLY_DD	Boolean	Holds a Master's degree
DOCTORATE_DD	Boolean	Holds a Doctorate degree

Field Name	Data Type	Description
WAR_COLLEGE_DD	Boolean	Holds a War College degree
SKIL_AQD_CD_1-20	String	Advanced Qualification Designation code
SKIL_SUBSPEC_CD_1-5	String	Subspecialty code
JOINT_QUAL_DD	Boolean	Joint Qualification completed
AQ_PROF_DD	Boolean	Advanced Qualification completed
IA_GSA_DD	Boolean	Individual Augmentee or Global War on Terrorism Support Assignment completed
JSO_DD	Boolean	Joint Service Officer position completed
JPME_I_II_DD	Boolean	Joint Professional Military Education I and II completed
JPME_I_ONLY_DD	Boolean	Join Professional Military Education I completed
LANG_ID_1-5	String	Foreign language code proficient

3. Data Exploration

In this section, we explored the data to determine their characteristics and potential usefulness as input variables. The following graphs show the initial insights gained from the data extracted.

Figure 27 shows the distribution of the aviation designator field. The data set consisted of 1,718 Naval Aviators (1310) and 832 Naval Flight officers (1320).

Figure 27. Distribution of Aviation Designators


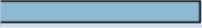
Value	Proportion	%	Count
1310.000		67.37	1718
1320.000		32.63	832

Figure 28 shows the distribution of ranks. The results showed that 68.71 percent of post-command aviators hold the rank of Captain (O7), while the next largest group, 24.16 percent, holds the rank of Commander (O5). Aviators usually assume command of a squadron at the rank of Commander, and therefore, rank could be used as a predictor.

Figure 28. Percentage of Ranks









Value	Proportion ▾	%	Count
O6		68.71	1752
O5		24.16	616
O8		2.71	69
O7		2.43	62
O9		1.37	35
O10		0.39	10
O4		0.12	3
O2		0.12	3

Figure 29 shows that 98.86 percent of post-command aviators are male, while only 1.14 percent are female. Gender could be used as an input variable.

Figure 29. Distribution of Males and Females


Value ▲	Proportion	%	Count
F		1.14	29
M		98.86	2521

Figure 30 shows the separation code that describe the reason for separation. The graph has missing data, indicated by the blank space, which shows that the aviator is active. Because these values do not offer any meaning, a node was created to translate each code.

Figure 30. Distribution of Separation Codes












Value	Proportion	%	Count
RBD		47.41	1209
		39.33	1003
SBC		9.8	250
RCC		1.8	46
SCC		0.59	15
RNC		0.47	12
DDD		0.24	6
SNC		0.12	3
SFJ		0.12	3
SFK		0.08	2
SGB		0.04	1

Figure 31 shows the distribution of advanced qualification designation (AQD) codes (OPNAV N13, 2015). There are 20 AQD fields for each record (SKIL_AQD_CD_1 – 20), and thus they had to be aggregated to translate the codes in to some form of distinguishable meaning. The missing fields were not a factor due to counts of the different types of AQDs across the twenty fields. AQDs are a measurable skill and could be used as an input variable.

Figure 31. Distribution of Advanced Qualification Designation Codes




















Value	Proportion	%	Count
DA7		14.9	380
DJ4		14.16	361
DV4		9.69	247
DD2		9.25	236
DL3		8.71	222
DB4		7.57	193
DV3		6.04	154
DW5		3.02	77
DWA		2.71	69
DD8		2.63	67
DK2		2.63	67
DF3		2.63	67
DB5		2.16	55
DF2		1.96	50
DW8		1.33	34
DA4		1.33	34
DJ3		1.29	33
DV1		1.29	33
DA5		1.1	28

Figure 32 shows the count of various subspecialty codes. Aviators obtain subspecialty codes based on their postgraduate field of study (OPNAV N13, 2015). Each

record has five subspecialty fields that required aggregation. This field also could be used as an input variable.

Figure 32. Distribution of Subspecialty Codes

Value	Proportion	%	Count
2000P		29.22	745
		13.14	335
3000P		6.9	176
5403Q		5.45	139
6301R		3.8	97
6301S		3.1	79
2000Q		3.1	79
3130S		2.12	54
\$null\$		2.12	54
3100P		1.73	44
6205R		1.49	38
5402P		1.18	30
3105P		0.98	25
4000P		0.94	24
6201P		0.9	23
5000P		0.9	23
3111Q		0.82	21
3110P		0.71	18
6205S		0.71	18

4. Data Quality

The data audit node revealed that only 17.5 percent of the fields were complete. Complete fields are listed in Table 2.

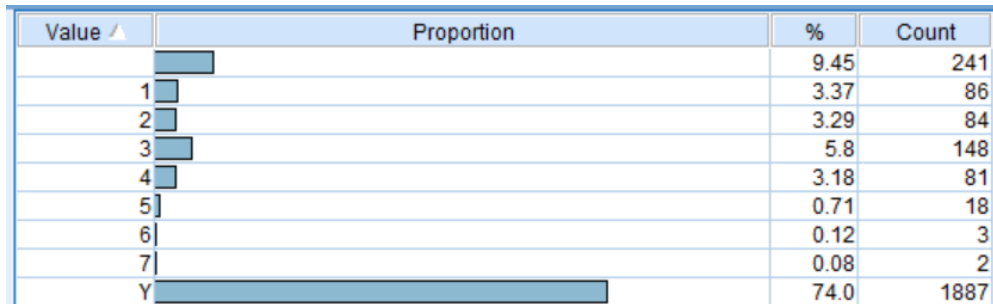
Table 2. Fields with Complete Values

Field Name
DESIGNATOR_DD
ACBD
ADSD
SRC_CD_PGM
CURRENT_GRADE
TOTAL_ACTIVE_SERVICE_DD
TOTAL_ACT_COMMISSION_SERV_DD
DOB
ETHIC_GRP_CD
RACE_CD
SEX_CD
MASTERS_ONLY_DD
SKIL_AQD_CD_1-4
PROG_PROM_HIST_DT_I_LCDR
PROG_PROM_HIST_H_CDR

The data audit also showed that no records were 100 percent complete. Many fields are empty because they are not normalized with several fields included for the same variable. The fields that contain the most null values are AQDs (SKIL_AQD_CD_1-20), NOBCs (ASGN_NOBC_1-12), and deployment duration (ASGN_DPLY_DURA1-12).

One example of errors in the data set is the dependents field shown in Figure 33. We see that there is a mix of numbers, a blank space, and a “Y,” which constitutes 74 percent of the entire field. Because the letter “Y” is ambiguous and represents the majority of the field, it could not be used as an input variable.

Figure 33. Distribution of the Number of Dependents per Household



We addressed missing values by considering fields that do not contain null values except in the deployment duration attribute. Figure 34 shows the different values that represent this field, where null represents no deployment time. Because 12 fields provide deployment time, the sum of these fields could be used as an input variable to determine whether deployment duration over a career affects retention. Null fields cannot be added and must be converted to “0” in order to correctly calculate the sum of the values in the set.

Figure 34. Distribution of the Deployment Duration

Value	Proportion	%	Count
\$null\$		96.94	2472
6.000		0.9	23
8.000		0.39	10
5.000		0.27	7
12.000		0.2	5
3.000		0.16	4
2.000		0.12	3
7.000		0.12	3
1.000		0.12	3
24.000		0.12	3
9.000		0.08	2
4.000		0.08	2
26.000		0.08	2
10.000		0.08	2
14.000		0.04	1
13.000		0.04	1
25.000		0.04	1
17.000		0.04	1
60.000		0.04	1
16.000		0.04	1
11.000		0.04	1
15.000		0.04	1
29.000		0.04	1

C. DATA PREPARATION

During this phase, we cleaned and transformed the data in preparation for analysis, and created new fields to improve our ability to interpret the data. The following shows the processes conducted to prepare the data. After transformation, we plotted the newly created input variables with the target variable as an overlay to see whether these variables have a predictive power of the target variable. The following sections discuss the main cleaning and transformation operations performed on the original data set.

1. Converting Timestamp Data Types to Integers

The original data set represented dates as timestamps. To calculate differences in years easily, we converted timestamps to four-digit integer years. Figure 35 shows the formula used to convert the ELG_RETIRE_DT field to a four-digit year. The original format is an integer of one or two digits, so we employed a derive node that uses a conditional formula to convert 50 or less to 2000 to 2050 and 51 and above to 1951–1999. This field was used to calculate time served after the service member was eligible to retire.

Figure 35. Year Eligible to Retire Formula

The screenshot shows a configuration window for a field named 'Year Eligible to Retire'. The window has two tabs: 'Settings' (selected) and 'Annotations'. The 'Mode' is set to 'Single'. The 'Derive field:' section contains the text 'Year Eligible to Retire'. The 'Derive as:' dropdown is set to 'Conditional'. The 'Field type:' dropdown is set to 'Continuous'. The 'If:' section contains the formula '1 ELG_RETIRE_DT < 50'. The 'Then:' section contains the formula '1 to_integer(ELG_RETIRE_DT + 2000)'. The 'Else:' section contains the formula '1 to_integer(ELG_RETIRE_DT + 1900)'. At the bottom, there are buttons for 'OK', 'Cancel', 'Apply', and 'Reset'.

Figure 36 shows the conversion of the retirement date to a four-digit year. All timestamped fields were in the YYYY-MM-DD HH:MM:SS format. First, we converted the field to a string, after which the first four characters of the string were extracted and then converted to an integer. As an integer, the field could now be used for calculations based on year differences. This process was repeated for the following fields: ACBD, ADSD, PROG_PROM_HIST_DT_I_LCDR, PROG_PROM_HIST_DT_H_CDR, and PROG_PROM_HIST_DT_G_CAPT.

Figure 36. Year Retired Formula

The screenshot shows a configuration window for a field named 'Year Retired'. The window has two tabs: 'Settings' (selected) and 'Annotations'. The 'Mode' is set to 'Single'. The 'Derive field:' section contains the text 'Year Retired'. The 'Derive as:' dropdown is set to 'Formula'. The 'Field type:' dropdown is set to 'Nominal'. The 'Formula:' section contains the formula '1 to_integer(startstring(4,to_string(SEPW_TRANS_LOSS_DT)))'. At the bottom, there are buttons for 'OK', 'Cancel', 'Apply', and 'Reset'.

In transforming timestamp fields, we constructed additional fields to store the newly created values. Table 3 shows the original and newly created field names.

Table 3. Fields Converted to Integers

Original Field Name	New Field Name
ACBD	Year_ACBD
ADSD	Year_ADSD
SEPN_TRNS_LOSS_DT	Year_Retired
PROG_PROM_HIST_DT_I_LCDR	Year_LCDR
PROG_PROM_HIST_DT_H_CDR	Year_CDR

2. Calculating Difference in Years

We used SPSS Modeler expressions to calculate difference in years. Table 4 shows the calculations of year differences between fields. The purpose of this was to determine the interval between certain career milestones.

Table 4. Expressions Used to Calculate Difference in Years

Expressions	New Field Name
Year_Retired - Year_Eligible_to_Retire	Active Past Eligible Retirement Date
Year_LCDR - Year_ACBD	Years_to_LCDR
Year_CDR - Year_ACBD	Years_to_CDR

3. Calculating Total Deployment Time

Deployment times had 12 fields. For better analysis, we summed all valid values to determine the total duration of deployment. As some fields had null values, conversion to zero was necessary to calculate the sum correctly. Figure 37 shows the way in which we used the filler node to convert all null and blank fields to the integer “0.” Figure 38 shows the calculation of deployment duration and the creation of a new field called Total time deployed.

Figure 37. Replacing Blank Values Expression

The dialog box has two tabs: 'Settings' (selected) and 'Annotations'. Under 'Fill in fields:', a list of fields is shown: ASGN_DPLY_DURA_10, ASGN_DPLY_DURA_11, ASGN_DPLY_DURA_12, ASGN_DPLY_DURA_2, ASGN_DPLY_DURA_3, ASGN_DPLY_DURA_4, and ASGN_DPLY_DURA_5. The 'Replace:' dropdown is set to 'Null values'. The 'Condition:' section contains a list with one item: '1 @BLANK(@FIELD)'. The 'Replace with:' section contains a list with one item: '1 0'. At the bottom are buttons for 'OK', 'Cancel', 'Apply', and 'Reset'.

Figure 38. Deployment Duration Expression

The dialog box has two tabs: 'Settings' (selected) and 'Annotations'. At the top, there is a 'Mode:' section with 'Single' selected and 'Multiple' unselected. The 'Derive field:' section contains a text box with 'Total time deployed'. The 'Derive as:' dropdown is set to 'Formula'. The 'Field type:' dropdown is set to 'Nominal'. The 'Formula:' section contains a list with one item: '1 ASGN_DPLY_DURA1 + ASGN_DPLY_DURA_2 + ASGN_DPLY_DURA_3 + ASGN_DPLY_DURA_4 + ASGN_DPLY_DURA_5 +'. At the bottom are buttons for 'OK', 'Cancel', 'Apply', and 'Reset'.

4. Converting Fields to Flags

Seven fields in the data set contained one-character values that indicated whether the record included an attribute. Figure 39 shows an example of a convert to flag operation performed on the Individual Augmentee (IA)/Global War on Terror Support Assignment (GSA) field. If the node found a “Y” in the IA/GSA field, it flagged it as true. Table 5 provides a full list of the fields converted.

Figure 39. Convert Flag Fields Example

The screenshot shows a software interface for configuring a field derivation. It has two tabs: 'Settings' (selected) and 'Annotations'. Under 'Settings', the 'Mode' is set to 'Single'. The 'Derive field:' text box contains 'IA/GSA_Assigned'. Below this, 'Derive as:' is set to 'Flag' and 'Field type:' is also set to 'Flag'. The 'True value:' is 'Yes' and the 'False value:' is 'No'. The 'True when:' section contains a rule: '1 IA_GSA_DD = "Y"'. At the bottom are four buttons: 'OK', 'Cancel', 'Apply', and 'Reset'.

Table 5. Fields Converted to Flags Names

Original Field Name	New Field Name
IA/GSA_DD	IA/GSA_Assigned
MASTERS_ONLY_DD	Masters_Degree
JPME I ONLY_DD	JPME I
JPME I AND II_DD	JPME I AND II
WAR_COLLEGE_DD	War_College
JOINT_QUAL_DD	Joint_Qualification
JSO_DD	Joint_Service_Officer

5. Code Mapping

Several fields included indiscernible codes. To make the values of these fields discernible and therefore easier to understand, they were translated from alphanumeric codes to names. Figure 40 shows one example of code mapping. Table 6 lists the other fields that we transformed similarly.

Figure 40. Map Codes Expression

Set field to	If this condition is true
NROTC	SRCE_CD_PGM = 5 or SRCE_CD_PGM = 4
Naval Academy	SRCE_CD_PGM = 1
Naval Flight Officer...	SRCE_CD_PGM = 38
Aviation Officer Ca...	SRCE_CD_PGM = 3
Aviation Reserve O...	SRCE_CD_PGM = 80
NESEP	SRCE_CD_PGM = 29

Table 6. Code Mapping Fields

Original Field Name	New Field Name
ETHNIC_GRP_CD	Ethnicity Code
RACE_CD	Race Code
SRCE_CD_PGM	Source Code
SEPN_SPD_CD	Separation Code
ASGN_NOBC_1_1-12	ASGN_NOBC_1_1-12_nobctype
SKIL_AQD_CD_1-20	SKIL_AQD_CD_1-20_aqdtype

6. Aggregation and Count of Attributes

There were two sets of fields that had to be aggregated to make the data useful for analysis. Figure 41 shows an example of the aggregation operation. Each record can have a maximum of five subspecialty codes listed; however, many are blank because an aviator generally has two or fewer assigned. By aggregating the five fields into a list, a count of non-null values yields each aviator's total number of subspecialty codes. This is represented as Count Total SSP, and could be used as an input variable. The other set of fields that we aggregated was LANG_ID_1-5.

Figure 41. Aggregation of Subspecialty Code Fields

The screenshot shows a software interface for creating a new derived field. At the top, there are tabs for 'Settings' and 'Annotations', with 'Annotations' selected. Below the tabs, there is a 'Mode' section with radio buttons for 'Single' (selected) and 'Multiple'. The 'Derive field:' section contains a text input field with the value 'Count Total SSP'. Below this, the 'Derive as:' dropdown menu is set to 'Formula'. The 'Field type:' dropdown menu is set to 'Nominal'. The 'Formula:' section contains a text area with the following formula: `count_non_nulls(['SKIL_SUBSPEC_CD_1', 'SKIL_SUBSPEC_CD_2', 'SKIL_SUBSPEC_CD_3', 'SKIL_SUBSPEC_CD_4', 'SKIL_SUBSPEC_CD_5'])`. At the bottom of the interface, there are buttons for 'OK', 'Cancel', 'Apply', and 'Reset'.

We also aggregated and counted attributes for the AQD and NOBC fields, but they required further classification. Figures 42, 43, 44, and 45 show examples of how this process was conducted. Due to the large number of unique NOBCs, we found that it was beneficial to categorize them into aviation and non-aviation NOBCs.

Figure 42. Mapping of NOBCs to a Category

Settings Annotations

Mode: ☐ Single ☒ Multiple

Derive from:

- ASGN_NOBC_1_1
- ASGN_NOBC_10_1
- ASGN_NOBC_11_1
- ASGN_NOBC_12_1
- ASGN_NOBC_2_1

Field name extension: Add as: ☒ Suffix ☐ Prefix

Derive as: TIP: Refer to selected fields by using @FIELD

Field type: ☒ Nominal ☐ Ordinal Default value:

Set field to	If this condition is true
Health	@FIELD > 0000 and @FIELD < 1000
Supply/Fiscal	@FIELD > 0999 and @FIELD < 2000
Science/Service	@FIELD > 1999 and @FIELD < 3000
Personnel	@FIELD > 2999 and @FIELD < 4000
Facilities	@FIELD > 3999 and @FIELD < 5000
Electronics Engin...	@FIELD > 4999 and @FIELD < 6000
Weapons Engine...	@FIELD > 5999 and @FIELD < 7000
Naval Engineering	@FIELD > 6999 and @FIELD < 8000
Aviation	@FIELD > 7999 and @FIELD < 9000
Naval Operations	@FIELD > 8999 and @FIELD <= 9999

OK Cancel Apply Reset

After mapping, NOBC fields were combined into a list and the aviation NOBCs were counted.

Figure 43. NOBC Aggregation and Count

Settings Annotations

Mode: ☒ Single ☐ Multiple

Derive field:

Count of Aviation NOBCs

Derive as:

Field type: ☒ Nominal ☐ Ordinal

Formula:

```
1 count_equal("Aviation", [ASGN_NOBC_1_1_nobctype, ASGN_NOBC_2_1_nobctype, ASGN_NOBC_3_1_nobctype, ASGN_NOBC_4_1_1])
```

OK Cancel Apply Reset

In additional, a total count of NOBCs was conducted by counting non-null values, thereby eliminating the issue of missing values.

Figure 44. Aggregation and Count of Total NOBCs

Settings Annotations

Mode: ☒ Single ☐ Multiple

Derive field:
Count of NOBCs

Derive as: Formula

Field type: Nominal

Formula:
1 count_non_nulls({ASGN_NOBC_1_1_nobctype,ASGN_NOBC_2_1_nobctype,ASGN_NOBC_3_1_nobctype,ASGN_NOBC_4_1_nobctype,ASGN_NOBC_5_1_nobctype,ASGN_NOBC_6_1_nobctype,ASGN_NOBC_7_1_nobctype,ASGN_NOBC_8_1_nobctype,ASGN_NOBC_9_1_nobctype,ASGN_NOBC_10_1_nobctype,ASGN_NOBC_11_1_nobctype,ASGN_NOBC_12_1_nobctype,ASGN_NOBC_13_1_nobctype,ASGN_NOBC_14_1_nobctype,ASGN_NOBC_15_1_nobctype,ASGN_NOBC_16_1_nobctype,ASGN_NOBC_17_1_nobctype,ASGN_NOBC_18_1_nobctype,ASGN_NOBC_19_1_nobctype,ASGN_NOBC_20_1_nobctype,ASGN_NOBC_21_1_nobctype,ASGN_NOBC_22_1_nobctype,ASGN_NOBC_23_1_nobctype,ASGN_NOBC_24_1_nobctype,ASGN_NOBC_25_1_nobctype,ASGN_NOBC_26_1_nobctype,ASGN_NOBC_27_1_nobctype,ASGN_NOBC_28_1_nobctype,ASGN_NOBC_29_1_nobctype,ASGN_NOBC_30_1_nobctype,ASGN_NOBC_31_1_nobctype,ASGN_NOBC_32_1_nobctype,ASGN_NOBC_33_1_nobctype,ASGN_NOBC_34_1_nobctype,ASGN_NOBC_35_1_nobctype,ASGN_NOBC_36_1_nobctype,ASGN_NOBC_37_1_nobctype,ASGN_NOBC_38_1_nobctype,ASGN_NOBC_39_1_nobctype,ASGN_NOBC_40_1_nobctype,ASGN_NOBC_41_1_nobctype,ASGN_NOBC_42_1_nobctype,ASGN_NOBC_43_1_nobctype,ASGN_NOBC_44_1_nobctype,ASGN_NOBC_45_1_nobctype,ASGN_NOBC_46_1_nobctype,ASGN_NOBC_47_1_nobctype,ASGN_NOBC_48_1_nobctype,ASGN_NOBC_49_1_nobctype,ASGN_NOBC_50_1_nobctype,ASGN_NOBC_51_1_nobctype,ASGN_NOBC_52_1_nobctype,ASGN_NOBC_53_1_nobctype,ASGN_NOBC_54_1_nobctype,ASGN_NOBC_55_1_nobctype,ASGN_NOBC_56_1_nobctype,ASGN_NOBC_57_1_nobctype,ASGN_NOBC_58_1_nobctype,ASGN_NOBC_59_1_nobctype,ASGN_NOBC_60_1_nobctype,ASGN_NOBC_61_1_nobctype,ASGN_NOBC_62_1_nobctype,ASGN_NOBC_63_1_nobctype,ASGN_NOBC_64_1_nobctype,ASGN_NOBC_65_1_nobctype,ASGN_NOBC_66_1_nobctype,ASGN_NOBC_67_1_nobctype,ASGN_NOBC_68_1_nobctype,ASGN_NOBC_69_1_nobctype,ASGN_NOBC_70_1_nobctype,ASGN_NOBC_71_1_nobctype,ASGN_NOBC_72_1_nobctype,ASGN_NOBC_73_1_nobctype,ASGN_NOBC_74_1_nobctype,ASGN_NOBC_75_1_nobctype,ASGN_NOBC_76_1_nobctype,ASGN_NOBC_77_1_nobctype,ASGN_NOBC_78_1_nobctype,ASGN_NOBC_79_1_nobctype,ASGN_NOBC_80_1_nobctype,ASGN_NOBC_81_1_nobctype,ASGN_NOBC_82_1_nobctype,ASGN_NOBC_83_1_nobctype,ASGN_NOBC_84_1_nobctype,ASGN_NOBC_85_1_nobctype,ASGN_NOBC_86_1_nobctype,ASGN_NOBC_87_1_nobctype,ASGN_NOBC_88_1_nobctype,ASGN_NOBC_89_1_nobctype,ASGN_NOBC_90_1_nobctype,ASGN_NOBC_91_1_nobctype,ASGN_NOBC_92_1_nobctype,ASGN_NOBC_93_1_nobctype,ASGN_NOBC_94_1_nobctype,ASGN_NOBC_95_1_nobctype,ASGN_NOBC_96_1_nobctype,ASGN_NOBC_97_1_nobctype,ASGN_NOBC_98_1_nobctype,ASGN_NOBC_99_1_nobctype,ASGN_NOBC_100_1_nobctype})

OK Cancel Apply Reset

To determine the count of non-aviation NOBCs, we created an expression to subtract aviation NOBCs from the total count.

Figure 45. Deriving Non-Aviation NOBCs

Settings Annotations

Mode: ☒ Single ☐ Multiple

Derive field:
Non Aviation NOBCs

Derive as: Formula

Field type: Nominal

Formula:
1 'Count of NOBCs' - 'Count of Aviation NOBCs'

OK Cancel Apply Reset

7. Defining Target Variable through Binning

The original data set did not contain a good target variable, so we had to create one. Binning a field creates a categorical variable based on its continuous values. Figure 46 shows the binning node. The bins generated converted the year fields into a nominal field, which was useful in showing the categories of years served. For example, by setting the number of bins to three, the binning node returned three categorical bins, 3-15, 15-27, and greater than 27. In the aviation community, the goal is for aviators to have completed a successful major command at 28 years of commissioned service, therefore a flag variable was the best candidate for the target variable for the model. Figure 47 shows the target variable derived by the creation of a flag field that indicated whether an aviator served more than 28 years.

Figure 46. Creating Bin Values

Settings Bin Values Annotations

Binning method: Fixed-width (No. of bins = 3)

Binned field: TOTAL_ACT_COMMISSION_SERV_DD

Tile:

Bins will be created using the values shown in the table

Bin	Lower	Upper
1	≥ 3	< 15.33333333
2	≥ 15.33333333	< 27.66666667
3	≥ 27.66666667	≤ 40

Read Values

OK Cancel Apply Reset

Figure 47. Deriving Target Variable

Settings Annotations

Mode: ☒ Single ☐ Multiple

Derive field:
28_years_of_service_or_more

Derive as: Flag

Field type: Flag

True value: T False value: F

True when:
1 TOTAL_ACT_COMMISSION_SERV_DD > 27

OK Cancel Apply Reset

8. Understanding New Variables

To understand how the values of the new variables relate to the target variable, all derived input variables were plotted with the target variable as an overlay to confirm their viability as predictor variables. For all graphs shown in this section, the red bar indicates personnel who had high retention, i.e., they served 28 years or more, while the blue indicates those who served fewer than 28 years. The X-axis for all graphs shows the number of aviators who had the attributes of the various input variables.

Figure 48 illustrates the relationship between the Individual Augmentee/Global War on Terrorism Support Assignment (IA/GSA) input variable and the target variable. The Y-axis shows whether the aviator completed an IA/GSA tour. The percentage of both input variables that affected the target were within ten percent, which does not provide a good indicator of retention.

Figure 48. Distribution of IA/GSA Assigned with Target Overlay

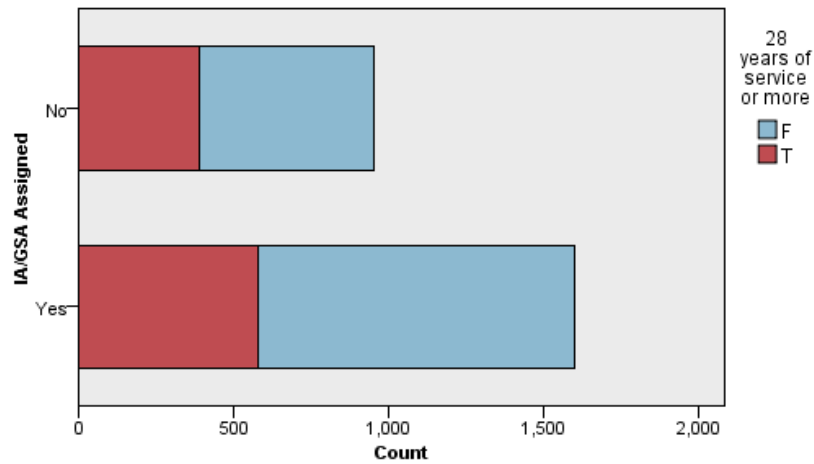


Figure 49 shows the relationship between aviators who had a joint qualification and the target variable. The Y-axis indicates whether the aviator had the joint qualification. The percentage of both input variable that affected the target were within ten percent, and thus, this also did not provide a good indicator of retention.

Figure 49. Distribution of Joint Qualification with Target Overlay

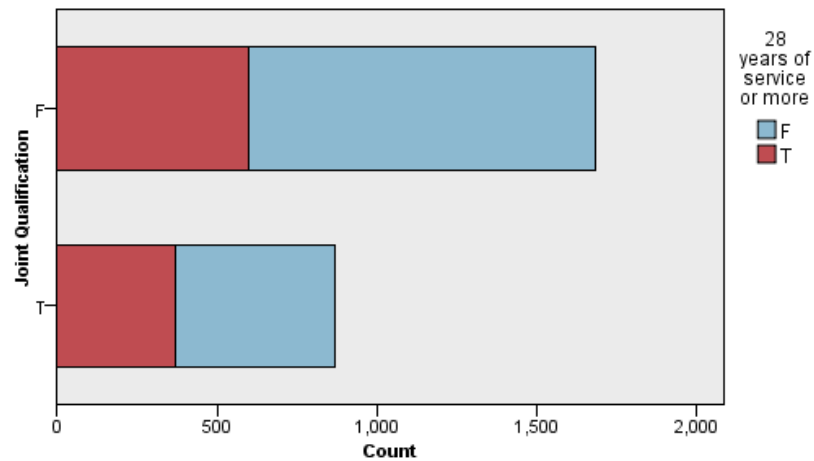


Figure 50 presents the relationship between Joint Service Officer (JSO) tour completion and the target variable. The Y-axis indicates whether the aviator served as a JSO. The percentage of both input variable that affected the target were within ten percent, and therefore, did not provide a good indicator of retention.

Figure 50. Distribution of Joint Service Officer Tour with Target Overlay

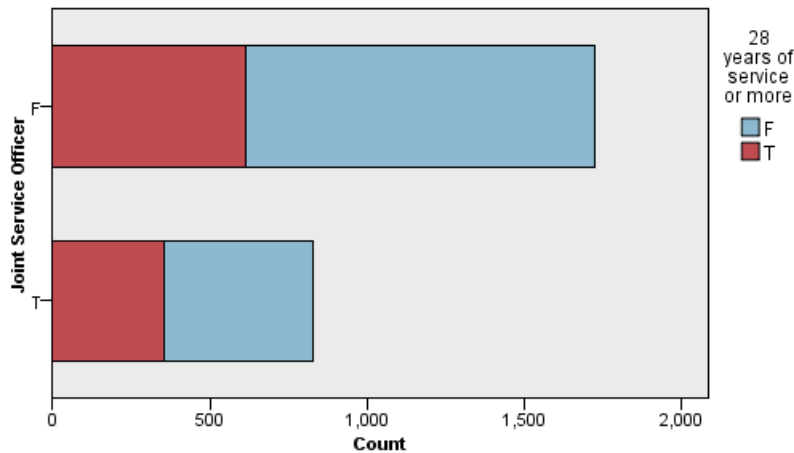


Figure 51 shows the relationship between an aviator who completed Joint Professional Military Education (JPME) Phase I and the target variable. The Y-axis indicates whether the aviator completed JPME Phase I. Those who completed JPME I have a 43 percent chance of leaving the service early compared to 23 percent who stayed for more than 28 years.

Figure 51. Distribution of JPME I Only Complete with Target Overlay

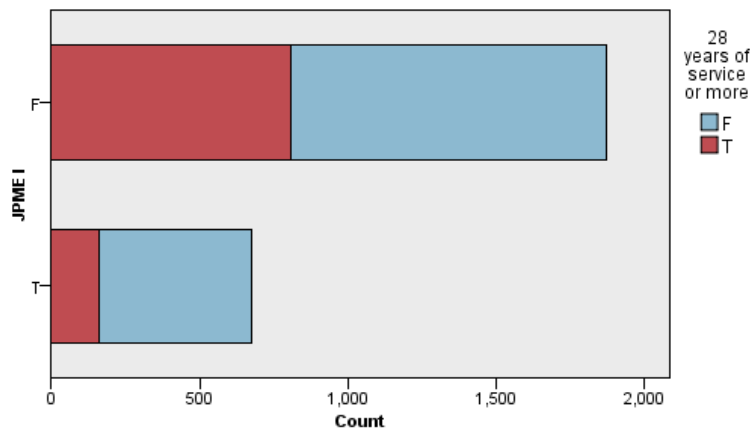


Figure 52 illustrates whether JPME I and II were completed. If the aviator completed JPME I and II, there was a 49 percent chance that s/he would remain in the service for 28 years compared to 36 percent for those who did not complete it.

Figure 52. Distribution of JPME I and II Complete with Target Overlay

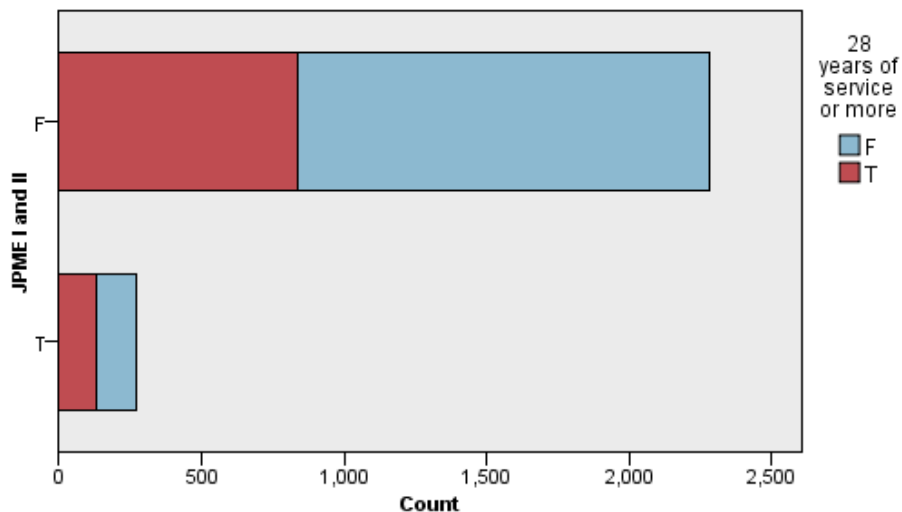


Figure 53 shows the relationship between aviators with a Master’s degree and the target variable. The percentage of both input variables that affected the target were within ten percent and did not provide a good indicator of retention.

Figure 53. Distribution of Master’s Degree with Target Overlay

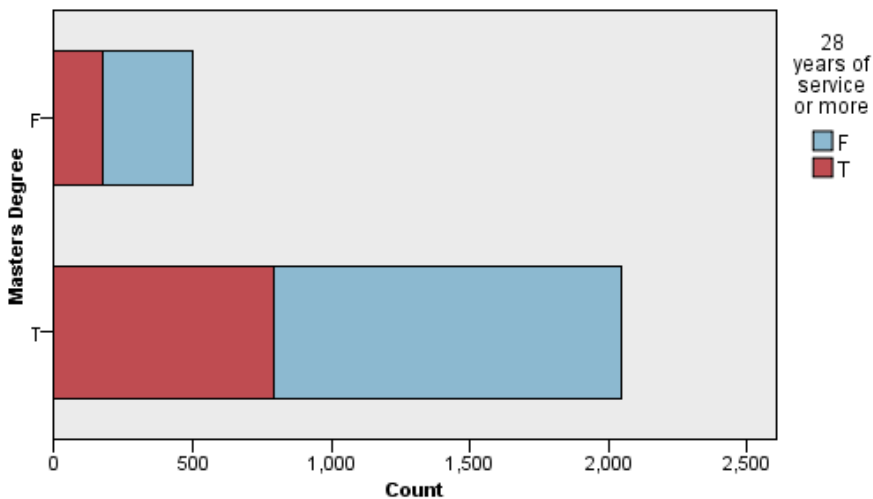


Figure 54 presents the relationship between aviators who attended war college and the target variable. The percentage of both input variables that affected the target were within ten percent and did not provide a good indicator of retention

Figure 54. Distribution of War College Education with Target Overlay

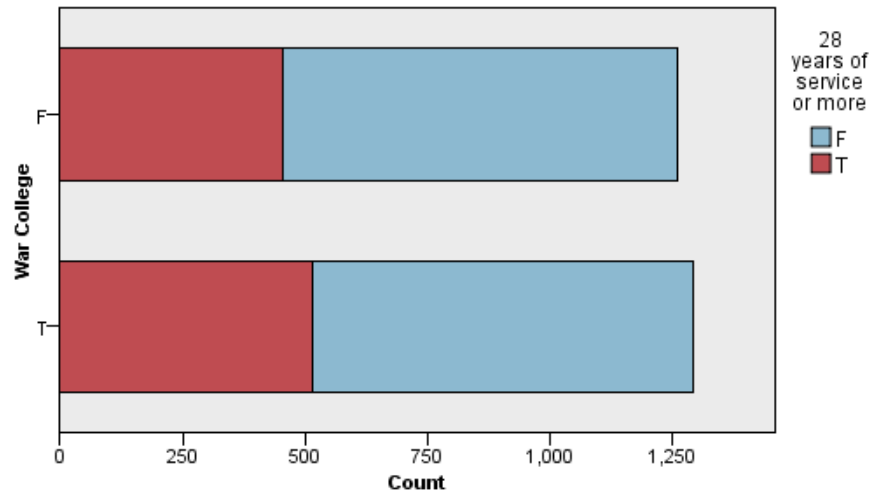


Figure 55 shows the relationship between aviators proficient in more than two languages and the target variable. If aviators know two or more languages, there is a 49 percent chance that they will serve for more than 28 years compared to 35 percent who only know one language, making this a good indicator of retention.

Figure 55. Distribution of Language Skills with Target Overlay

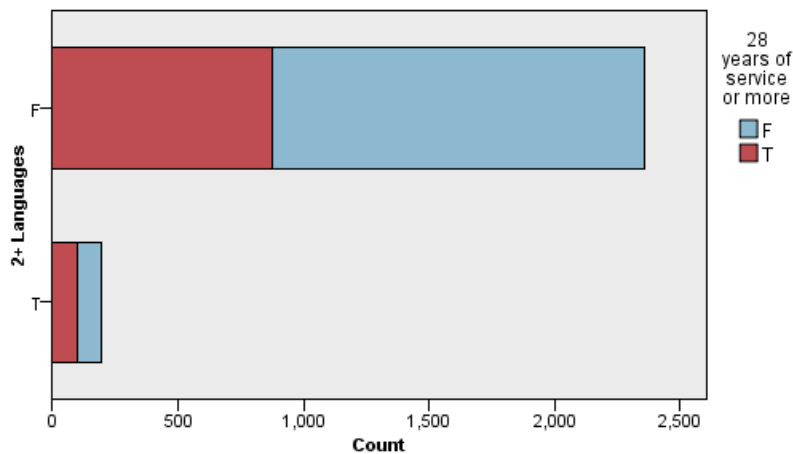


Figure 56 illustrates the relationship between an aviator's commissioning source and the target variable. The Y-axis illustrates the different categories of commissioning sources. Aviators who commissioned as an Aviation Officer Candidate, Naval Flight

Officer Candidate, Aviation Reserve Officer, or from the Navy Enlisted Science Education program (NESEP) programs were most likely to serve more than 28 years.

Figure 56. Distribution of Source Code with Target Overlay

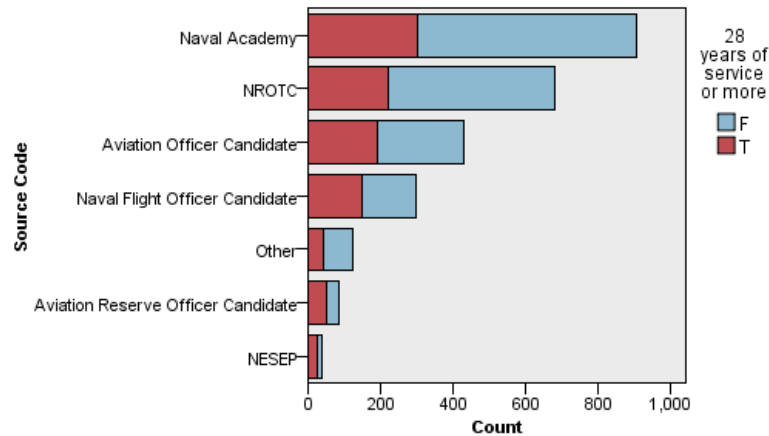


Figure 57 shows the relationship between the number of subspecialty codes an aviator obtained and the target variable. If an aviator has three or more subspecialty codes, there is a 50 percent chance that they will serve for 28 years or more, compared to 40 percent who have two or fewer subspecialty codes which is a good indicator of retention.

Figure 57. Distribution of Count of Subspecialty Codes with Target Overlay

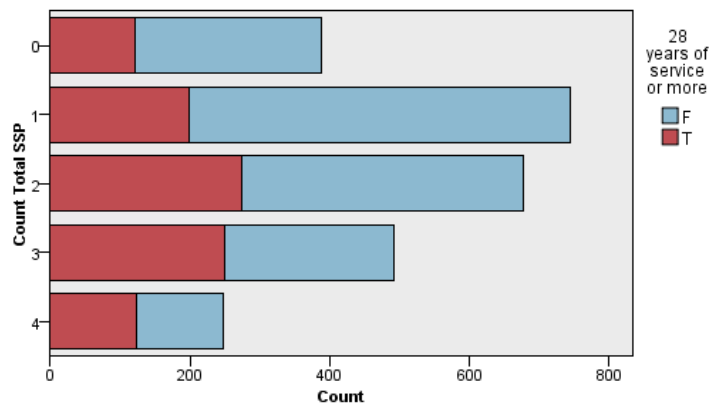


Figure 58 presents the relationship between deployment duration and the target. If the target has more than 20 weeks of deployment recorded, there is a 50 percent chance that s/he will serve for 28 years, compared to 33 percent if they have less than 20 weeks of deployment. Thus, this is a good indicator of retention.

Figure 58. Distribution of Deployment Duration with Target Overlay

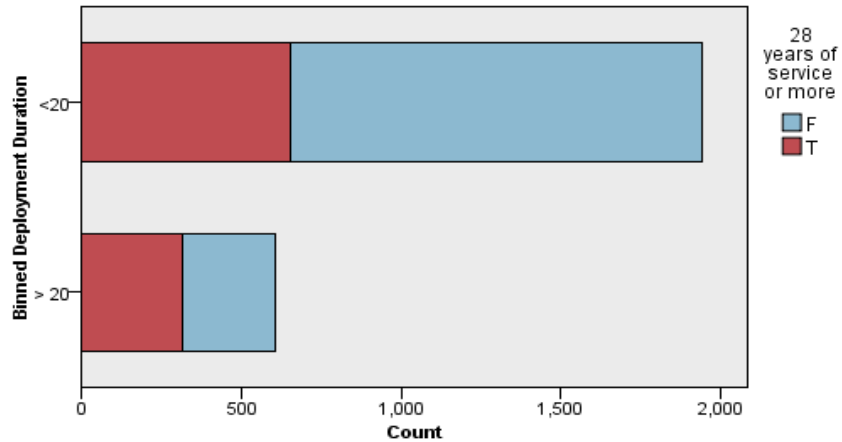


Figure 59 illustrates the relationship between the total count of AQDs an aviator has and the target variable. There was a similar correlation with the target variable across the different counts of AQDs. The same pattern applied to the counts of aviation and non-aviation AQDs. Therefore, this was not a good indicator of the target variable.

Figure 59. Distribution of Count of AQDs with Target Overlay

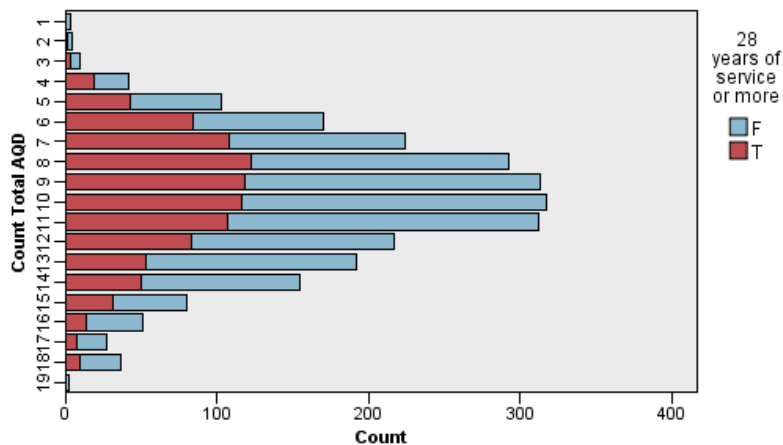


Figure 60 shows the relationship between the total number of NOBCs and the target variable. With eight and nine NOBCs, there is a 20 percent increase in the number of aviators who served more than 28 years. This is reasonable, because the longer the career, the more NOBCs are accumulated, but may not be a good predictor for a model that is applied early in the aviator's career.

Figure 60. Distribution of Count of Total NOBCs with Target Overlay

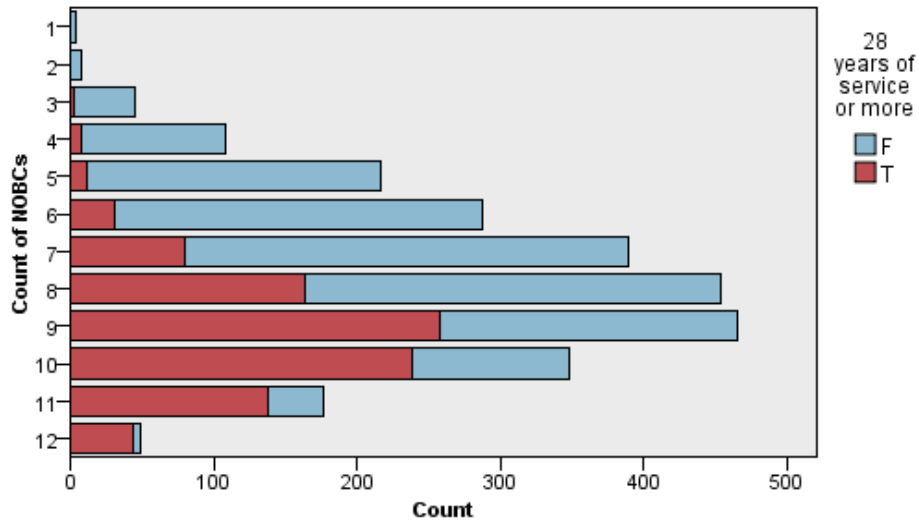


Figure 61 shows the relationship between aviation-specific NOBCs and the target variable; there was no correlation between the two. Therefore, this was not a good indicator of the target variable.

Figure 61. Distribution of Count of Aviation NOBCs with Target Overlay

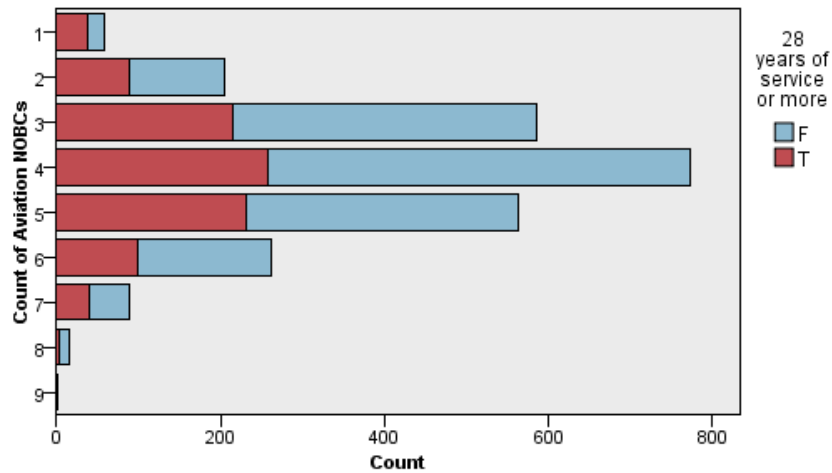
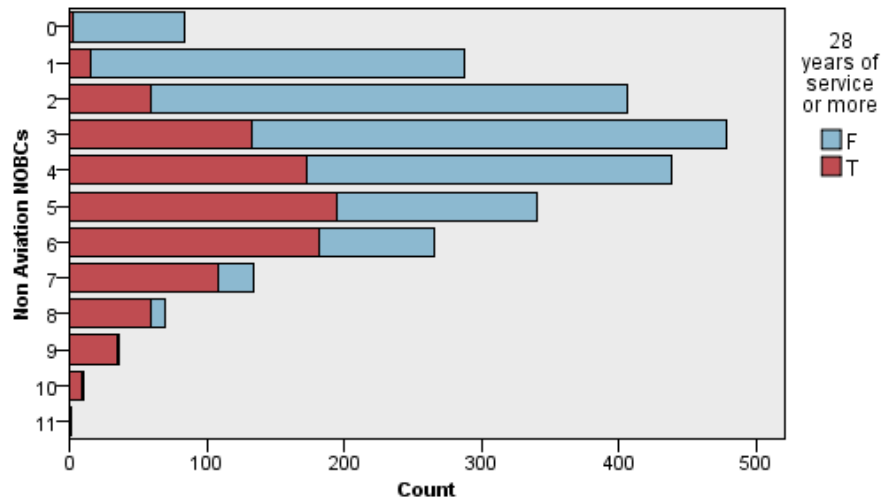


Figure 62 indicates the relationship between the number of non-aviation NOBCs and the target variable: when aviators have four and five non-aviation NOBCs, there is a 20 percent increase in the number who served more than 28 years. This could be a potentially good predictor of the target variable, particularly if these NOBCs are acquired early in an aviator's career.

Figure 62. Distribution of Count of Non-Aviation NOBCs with Target Overlay



D. MODELING

1. Selecting the Model

The first step in building a model is identifying the input and target variables. Table 7 lists the input and target variables selected. The target was a post-command aviator that has served for 28 years or more. This is because they had served long enough to be promoted to the rank of Captain and screened for major command and therefore indicative of successful retention.

Table 7. Input and Target Variables

Field Name	Description	Measurement Type	Role
SEX_CD	Provides gender	Flag	Input
Years to CAPT	Provides number of years to make CAPT	Nominal	Input
Years to CDR	Provides number of years to make CDR	Nominal	Input
Years to LCDR	Provides number of years to make LCDR	Nominal	Input
Master's Degree	Has Master's Degree	Flag	Input
War College	Has attended a War College	Flag	Input
JPME I	Has completed JPME Phase I only	Flag	Input
JPME I and II	Has completed JPME Phase I and II	Flag	Input
Joint Qualification	Is qualified in Joint operations	Flag	Input
Joint Service Officer	Is qualified as a Joint Service Officer	Flag	Input
Source Code	Source of commissioning	Nominal	Input
Binned Deployment Duration	Has total deployment time of 20 months or less	Flag	Input
Count of NOBCs	Total number of NOBCs	Nominal	Input
Non-aviation NOBCs	Number of non-aviation NOBCs	Nominal	Input
Count of Aviation NOBCs	Number of aviation specific NOBCs	Nominal	Input
Count Total AQDs	Total number of AQDs	Nominal	Input

Field Name	Description	Measurement Type	Role
Count of Aviation Warfare AQDs	Number of aviation specific AQDs	Nominal	Input
Count Non Aviation AQDs	Number of non-aviation AQDs	Nominal	Input
Count of Languages	Number of languages in which a member is proficient	Nominal	Input
2+ Languages	Has 2 or more languages	Nominal	Input
Count Total SSP	Total number of subspecialty codes	Nominal	Input
28 years of service or more	Has served more than 28 years	Flag	Target

The models we preferred to use were binary, as they provide a simple and understandable representation. The two models generated were the Quick, Unbiased, Efficient Statistical Tree (QUEST), and the Classification and Regression (C&R) Tree. The C&R tree creates levels with binary outputs by partitioning the training data set recursively. C&R trees are useful for prediction and classification. QUEST allows quicker processing than does C&R trees and reduces the inclination to use inputs that create more forks (Loh & Shih, 1997; IBM, 2010).

2. Generating Test Design

The data set consisted of 2550 records, which is relatively small; therefore, partitioning the data into training and test sets was not used as it would not likely provide higher quality results. Therefore, the model was generated from the entire data set.

3. Building the Model

Using SPSS as the modeling tool, we generated the binary models selected. Of the 1,816 records, 676 aviators in the data set reached the target variable, while 1,140 served less than 28 years. Figure 63 shows the QUEST model. The first binary split was on the number of non-aviation NOBCs. Aviators with five or more NOBCs were more likely to serve 28 years or more (67 percent) than those with fewer than five non-aviation NOBCs (33 percent). The next level of the tree indicates that the Years_to_LCDR was the next

important predictor of meeting the target. If the aviator made the rank of Lieutenant Commander at 8, 9, or 11 years of commissioned service (YCS), then there was an 80 percent chance that s/he would serve more than 28 years; if an aviator was promoted to Lieutenant Commander after the normal ten YCS, they had only a 57 percent of serving 28 years or more. At eight or nine YCS, aviators are considered below-zone and are promoted early. Those who are at 11 YCS are above-zone and were promoted at their last opportunity. For aviators who were promoted below or above-zone, we assumed that they desired to remain in the service. The model subdivided the ten-year mark to promote to LCDR further into the total NOBCs, among which, 59 percent were predicted to serve more than 28 years.

Figure 63. QUEST Model Decision Tree

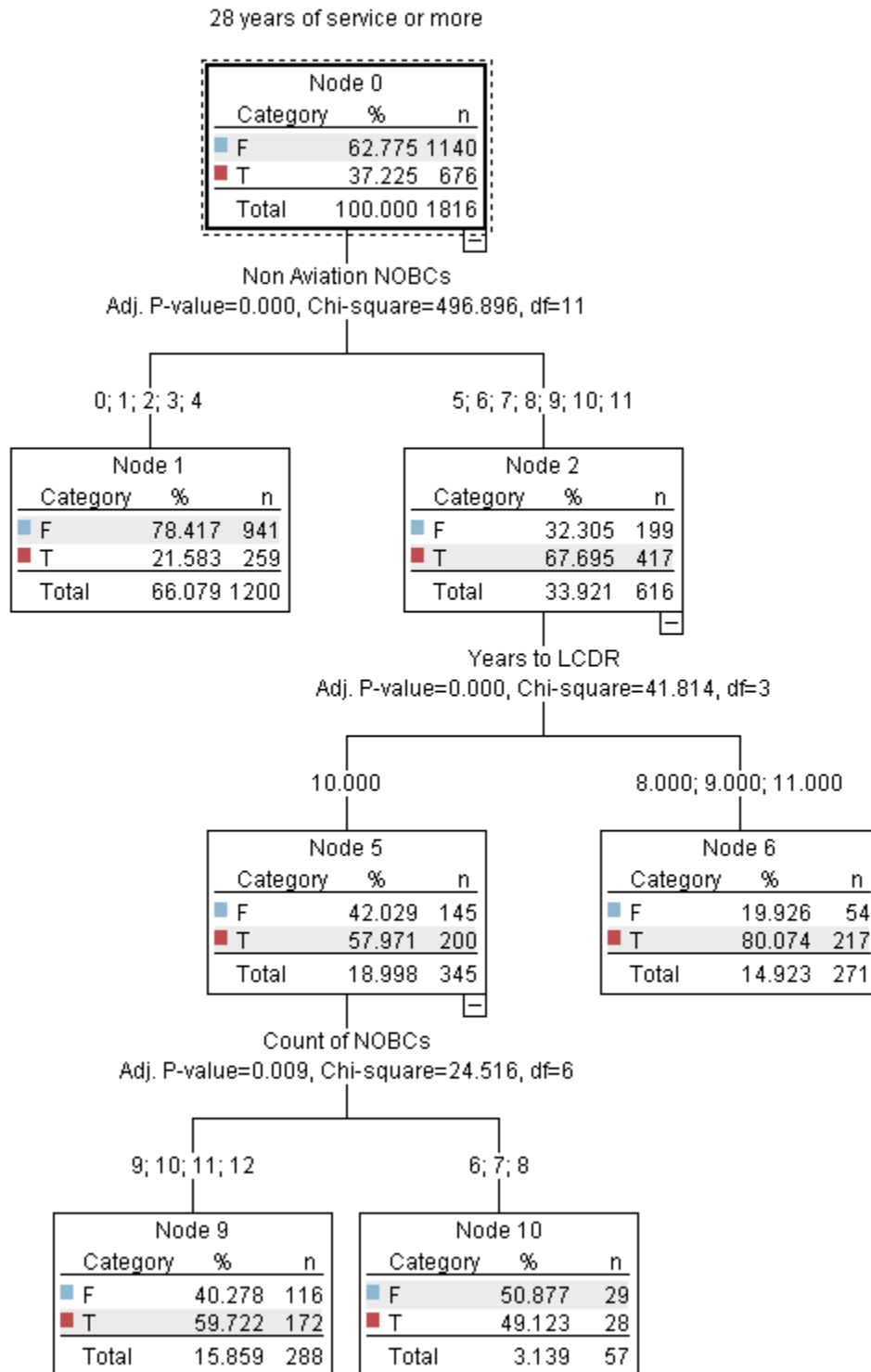
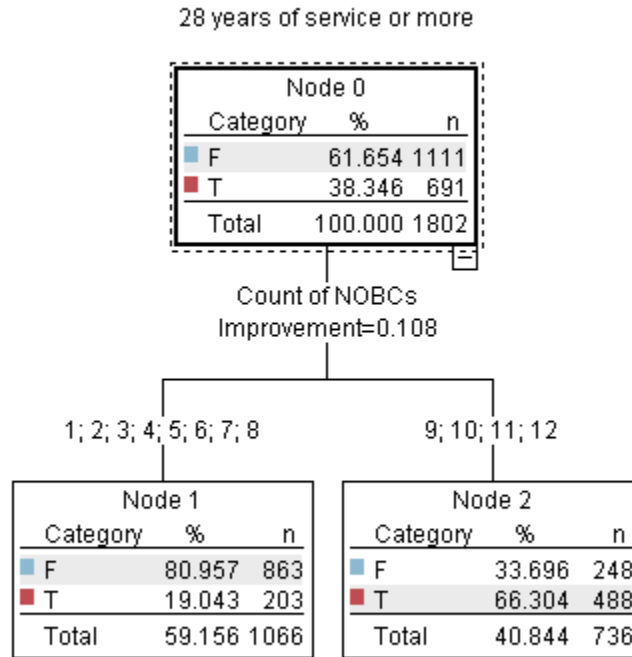


Figure 64 shows the C&R tree model, which had only one level for the binary split, the Count_of_NOBCs. Of the aviators who had nine or more NOBCs, 66 percent of them are likely to serve 28 years or more compared to 19 percent with eight or fewer NOBCs.

Figure 64. C & R Decision Tree



4. Assessing the Model

Figures 65 and 66 show that the overall accuracy of both models was 74 percent. After running the model nodes on the data set, a new field, \$XF-28_years_of_service_or_more, was created. We measured its accuracy by comparing the actual 28_year_of_service_or_more and the newly generated field.

Figure 65. Model Summary

Use?	Graph	Model	Build Time (mins)	Max Profit	Max Profit Occurs in (%)	Lift (Top 30%)	Overall Accuracy (%)	No. Fields Used	Area Under Curve
<input checked="" type="checkbox"/>		Quest 1	< 1	1617.805		30 1.869	74.706		10 0.734
<input checked="" type="checkbox"/>		C&R Tree 1	< 1	1546.635		40 1.712	74.275		6 0.734

Figure 66. Model Accuracy

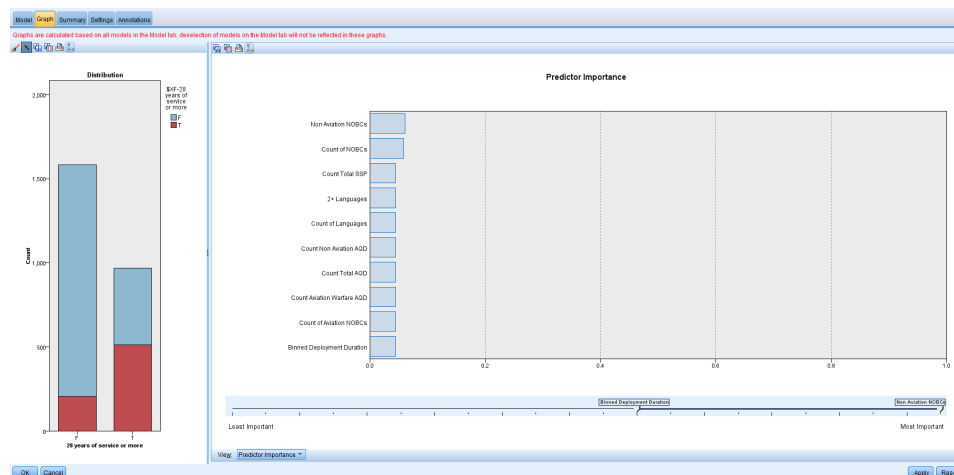
Results for output field 28 years of service or more

Comparing \$XF-28 years of service or more with 28 years of service or more

Correct	1,888	74.04%
Wrong	662	25.96%
Total	2,550	

Figure 67 shows that the predictor important to the models lay primarily in total NOBCs and non-aviation NOBCs. It appeared that non-aviation NOBCs were a better indicator because the likelihood of having five is greater at lower ranks. This is true with respect to business objectives, as the aviation community retains those who have diverse careers.

Figure 67. Predictor Importance



E. EVALUATION

1. Evaluating the Results

After reviewing the models and examining the results of the predictive set, we found that the business objectives are not truly met. The research found one indicator of retention, non-aviation NOBCs; however, we believe that the model can be improved with access to a more diverse set of data and by performing the analysis on a particular timeframe. The use of the QUEST model is preferred because it provides a more in-depth

analysis by showing more branches and hence more variables that contribute to the prediction of post-command aviator career retention.

2. Reviewing the Process

Assessment of the models showed that non-aviation NOBCs were the most important predictors of the target. For this model to be effective, the data set would have to be analyzed at a specific point in time. For example, if we wanted to analyze an Admiral's record, the comparison should be conducted on their record at the LCDR rank. The testing set would be active Lieutenant Commanders, and thus the target needs to match the testing set.

3. Determining the Next Steps

To fulfill the business objectives, the data set needs to include more sources. Using SPSS Modeler as a predictive modeling tool may not be the optimal way to incorporate all the data required to analyze the data and create a model to address the business objective. The research should consider moving on to a Big Data architecture, which facilitates the use of larger and diverse data sets and the ingestion of multiple sources into workable predictive analytics architecture.

F. DEPLOYMENT

The deployment phase includes the deployment plan, maintenance and monitoring, and development of the final report. Because the data set and model were insufficient, this model should not be deployed in its current state. In the next chapter, we address the further requirements that will make this a more accurate model.

G. CHAPTER SUMMARY

This chapter illustrated the step by step process of the CRISP-DM method. We discussed business understanding, data understanding, data preparation, modeling, evaluation, and deployment as they applied to the dilemma of Navy MPTE post-command aviator retention. The next chapter presents our summary, conclusions, and recommendations.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. SUMMARY, CONCLUSION, AND RECOMMENDATIONS

This thesis addressed two main issues: examining and proposing an end-to-end application architecture for performing analytics for Navy MPTE and determining whether a model could be developed to predict retention of post-command aviators. The purpose of this chapter is to: 1) provide a summary of our research effort on proposing an end-to-end application architecture for MPTE and developing a predictive model for post-command aviator retention using data mining techniques, 2) address the research questions posed in Chapter I, and 3) provide recommendations for future research as well as some final concluding thoughts.

A. SUMMARY

First, we examined an end-to-end application architecture for performing analytics for Navy MPTE. In Chapter III, we looked at a traditional architecture based on a data warehousing approach. Data warehouse architectures can be classified into four different types: enterprise data warehouse, operational data store, data vaults, and hub and spoke data marts. Data warehouses are efficient in analyzing relatively small, structured data, but fall short in analyzing unstructured, large data sets that are greater than Petabytes in size. With the exponential collection and growth of diverse data in the recent years, data warehouses will not be able to support the storage and analysis of this data.

To address this issue, we examined a Big Data architecture and its main components. We first looked at the Data Source Layer and identified that data can either be structured, semi-structured, or unstructured. Then we examined the Ingestion Layer to see how different types of data can be integrated in a Big Data architecture, and the different ingestion tools that could be used. We looked next at the Storage Layer, which stores the data for analysis and examined the advantages and disadvantages of different technologies for the storage layer of Big Data. Next, we looked at different tools and technologies such as MapReduce, Impala, Pig, and Hive that are part of the Hadoop Platform Management Layer and Analytics Engine. We then examined the Visualization

Layer which includes different techniques and tools for data visualization on either data warehouses or Big Data. Lastly, we examined the Security and Monitoring Layer and the different techniques that are implemented to protect information and provide a secure environment.

Second, this thesis addressed whether a model can be built to predict retention for post-command aviators. The methodology used for this project was the CRISP-DM process, a well-known framework that represents how data mining projects are done in the real world (Chapman et al., 2000). Each phase of the CRISP-DM process and its respective tasks were presented and discussed in Chapter IV. The first phase of CRISP-DM, Business Understanding, was conducted in Chapter II in which we examined Navy MPTE's organizations, current systems and processes, and the Talent Management Initiatives. Additionally, we examined several challenges that Navy MPTE is experiencing, focusing on retention of the post-command aviation community. In Chapter V, we determined the business objectives and the data mining goals of the project. Navy MPTE's business objective was to improve retention of the post-command aviator. The data mining goal was to build a predictive model for post-command aviators and identify significant indicators that will help in predicting retention. In Data Understanding, we collected officer billet, training, and education data from the NMPBS. We then explored the various fields and determined whether the data in its raw format was sufficient for analysis or whether further manipulations were required. Data Preparation involved cleaning, transforming, and constructing fields based on the original data set. In the Modeling phase, we selected two binary decision trees to develop the predicting model in order to address the business objectives and data mining goals. In the Evaluation phase, we determined that more data sources were needed for the models to be effective. Deployment of the current model is not recommended until more data sources are included in the analysis.

B. REVIEW OF RESEARCH QUESTIONS

The main objectives of this research were to examine and propose an end-to-end application architecture for performing analytics for Navy MPTE, and to develop a

predictive model for retention to address Navy MPTE's concerns regarding retention of the post-command aviators. The following is a recap of our research questions posed in Chapter I and how they were addressed by the effort of this thesis.

1. Data Architecture Questions

- What are the various internal and external data sets that need to be analyzed?

This thesis mainly focused on internal data sets of MPTE, such as OPINS and NES for officer and enlisted data, NTMPS for training data, NSIPS for pay and personnel information, and NMPBS for historical records. External data sources such as social media were beyond the scope of this thesis as access was not available to those data sets. Our recommendations on how external data sets can be incorporated is covered in the Recommendations Section.

- How is the ingestion of the data into the Hadoop environment accomplished from the data sources?

With the majority Navy MPTE's databases classified as relational databases, the best tool for ingestion into a Big Data platform is Sqoop, which is specifically designed to support the ingestion of data from RDBMS to HDFS (Teller, 2015). Sqoop automates most of the ingestion process and can support incremental imports of data, only retrieving records newer than the previously imported set ("Sqoop User Guide," 2016). This feature would be helpful to Navy MPTE as their databases update monthly.

- What are the necessary Hadoop infrastructure hardware and software components?

The two infrastructures that can support Big Data are physical and cloud computing. Physical infrastructures are owned and maintained by the organization, while third-party vendors provide the infrastructure for cloud computing. In recent years, the Department of Defense (DOD) has pushed towards establishing the DOD Cloud Environment in order to reduce the current IT infrastructure (DOD, 2012). In line with the DOD Cloud Computing Strategy, our recommendation is to use cloud computing as the underlying infrastructure for a Big Data architecture. According to the DOD (2012), the benefits of cloud computing include improved server utilization, immediate increase

or decrease of servers, the ability to take advantage of emerging technologies of the private sector, and a shift from managing and maintaining hardware to only managing services.

Many analytic tools are available that can support Navy MPTE's mission by creating a common operating picture. Impala and Mahout are good candidates that can provide the necessary capabilities to meet the requirements of Navy MPTE. Impala provides the capability to process data of different types and formats, including text, efficiently. Mahout is a machine learning library of algorithms that can analyze data sets of different sizes to create predictive and other models.

- What are the different types of NoSQL databases that are most suitable to store Navy MPTE data?

Based on the current data sources, we theorize that the most suitable NoSQL database to meet Navy MPTE's business needs is a graph data store. This type of database provides for the analysis of complex relationships of data, and is suited for recommendation systems (Hecht & Jablonski, 2011). Additionally, a column-oriented database, like HBase, can be used to store data from social media (Sawant & Shah, 2013).

- Does Navy MPTE need Big Data technology or should it instead use a high-performing, relational database management system (RDBMS) and traditional Data Warehouse technology?

The overall architecture we recommend for Navy MPTE is a combination of Big Data and traditional relational database management systems. Depending on the type of analysis Navy MPTE requires, they can take advantage of what both Big Data and RDBMS has to offer. RDBMS allows for the inserting, updating, and deleting of records, while Big Data processing tools like Hive and Impala does not. In addition, RDBMS are beneficial for Navy MPTE for analyzing small data sets and providing immediate results (Cloudera, 2014). However, Big Data analytics and processing tools like Pig, Hive, and Impala are optimized for large amounts of data, can support complex data types, and scale at relatively low cost (Cloudera, 2014).

Analyzing data is a challenging task as Navy personnel data is distributed across multiple databases. A Big Data architecture provides a way to integrate multiple data sources without needing to replace current RDBMS. In addition, a Big Data architecture takes advantage of NoSQL databases in storing structured and unstructured data, allowing for a more efficient way of organizing data vice having multiple records for a single person in a relational database, which does not scale well when data grows.

2. Data Mining Project Questions

- What are the substantive issues that a Navy MPTE Common Operating Picture is trying to solve?

Navy MPTE's motivation for developing a Common Operating Picture is to identify and retain the most talented Sailors. There are several factors that can be used to identify the most qualified Sailors such as their educational level, training, billet history, and FITREPs. Currently, the Navy is losing a good amount of experienced and talented officers to the private sector, which is an important concern for Navy leadership and the health of the fleet in the future.

- What aviation talent is being lost?

This question was difficult to answer due to insufficient information in the current data set. To accurately answer this question FITREP data and other external data sources would be needed to allow us to quantify a Sailor's talent.

- What are some indicators that would lead an Officer to leave the service?

Navy Officer Billet Classifications (NOBC) were discovered as the most important indicator to determine whether an officer leaves the service. Aviators are likely to leave the service if they have eight or fewer NOBCs.

- Can a model be developed from available data to predict post-command aviator retention?

We believe it is possible to develop a model for predicting retention of post command aviators with reasonable accuracy using available techniques. However, with our limited data set, the model we developed has a limited accuracy. For a more accurate model, additional data sources are needed for developing and validating the model.

C. RECOMMENDATIONS

The following are our recommendations for future research.

1. Other Internal Databases and Fitness Reports/Enlisted Evaluations

This thesis was limited to only using OPINS as the primary source of data. Other internal databases can significantly improve the accuracy of the developed predictive model. Training data, located in NTMPS, can pinpoint what talent is being retained or lost. NES can also be incorporated to predict retention for enlisted Sailors.

One important indicator that was not included in this research was FITREPs/Enlisted Evaluations. FITREPs/Enlisted Evaluations contain information regarding billets, performance trait average, physical fitness scores, and milestone recommendations of the reporting senior. This information can be found in relational databases like NSIPS. However, one important piece of information that is not being captured is the reporting senior's comments on the performance of the officer, which is in text format. These performance comments are used to generate soft breakouts of the Department Head rankings, and by aviation command screen boards to fully capture and understand an officer's performance as trait averages tell half the story.

There are several text mining tools such as IBM's Advanced Text Analytics Toolkit capable of capturing the performance comments in FITREPs. Through text analytics, applications can extract keywords or phrases from unstructured text and derive structured data from it (Zikopoulos et al., 2012). This additional information could be used to develop better and more accurate predictive models.

2. External Data Sources

This research mainly focused on internal data sources for Navy MPTE, but Big Data architectures can incorporate external data sources as well which may include social media and employment data. Social media applications have completely changed the way people interact with each other. People are posting anything from their opinions about issues, their likes and dislikes, lifestyle changes, and their plans and future decisions. Social media is becoming an effective tool in capturing and summarizing a person's life.

Employment data can be useful to Navy MPTE as they can gain an understanding of the current trends in the job market.

Sentiment analysis using text analytics can be applied to social media in order to gain insights into Sailors' opinions regarding the Navy and employment opportunities outside the Navy (Sawant & Shah, 2013). Additionally, sentiment analysis can help in improving work conditions and the retention rate of the Navy.

3. Security and Privacy

Security and privacy are an important aspect of any organization that collects data. The amount and variety of the collected data makes it an important asset for any organization. This information could contain detailed sensitive data regarding employees' health, financial situation, personal life, and family information. Hackers and cyber criminals look for this type of information to exploit and target service members and their families. The organization must establish strict and secure policies to protect their employee information. These policies must also comply with the federal laws establish to protect the privacy and security of the information set by the government.

D. CONCLUSION

Prior to undertaking a data mining project, it is important to first understand the business objectives and ensure a data architecture that is in place can support it. Big data architectures are an emerging technology but is not meant as a complete replacement for traditional relational databases. A robust data warehouse with business intelligence tools is sufficient for analyzing small data sets. Once a data architecture is selected, the data mining project can begin. Frameworks, like the CRISP-DM process, must be used in order to build an efficient model; without it, data mining projects will likely to fail.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- Abbott, D. (2014). *Applied predictive analytics: Principles and techniques for the professional data analyst*. Indianapolis, IN: John Wiley and Sons.
- Amazon. Amazon Elastic MapReduce: Developer guide. (2015). Retrieved from <http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-what-is-emr.html>
- Apache. (n.d.-a). Apache Spark [Information on page]. Retrieved December 2015, from <https://spark.apache.org/>
- Apache. (n.d.-b). Apache Storm [Information on page]. Retrieved December 2015, from <https://storm.apache.org/>
- Apache. (2012). Flume 1.6.0 user guide [Information on page]. Retrieved from <https://flume.apache.org/FlumeUserGuide.html>
- Apache. (2013). Hadoop MapReduce [Information on page]. Retrieved from https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html#Overview
- Apache. (2014a). Apache Hive [Information on page]. Retrieved from <https://hive.apache.org/>
- Apache. (2014b) Zookeeper [Information on page]. Retrieved from <http://zookeeper.apache.org/doc/trunk/zookeeperOver.html>
- Apache. (2016a). Apache Lucene [Information on page]. Retrieved from <http://lucene.apache.org/>
- Apache. (2016b). Sqoop user guide (v.1.4.2) [Information on page]. Retrieved from https://sqoop.apache.org/docs/1.4.2/SqoopUserGuide.html#_introduction
- Borthakur, D. (2008). HDFS architecture guide. Retrieved from https://hadoop.apache.org/docs/r1.2.1/hdfs_design.pdf
- Bureau of Naval Personnel. (2002, Sep 26). *Mission and functions of Navy Personnel Command (NAVPERSCOM)* (BUPERS Instruction 5450.54). Millington, TN: Chief of Naval Personnel.
- Chandran, S. S. (2014). *Making better business decisions with analytics and business rules: An overview of the IBM decision management product stack*. Retrieved from http://www.ibm.com/developerworks/bpm/library/techarticles/1407_chandran/1407_chandran-pdf.pdf

- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. Retrieved from <https://www.the-modeling-agency.com/crisp-dm.pdf>
- Cloudera. (2016). Cloudera Impala guide [Information on page]. Retrieved from <https://www.cloudera.com/documentation/enterprise/5-3-x/topics/impala.html>
- Cloudera. (2014). Choosing the best tool for the job [Lecture notes]. Retrieved from <https://cle.nps.edu/access/content/group/a20708ea-7b82-4441-9119-97ad0b401f77/Lecture%20Presentations/Week%2010%20Lectures.pdf>
- Cloudera. (2013). Introduction to data science: Building recommender systems [Lecture notes]. Retrieved from <http://training.cloudera.com>
- CSC. (2012). Big data universe beginning to explode [Information on page]. Retrieved from http://www.csc.com/insights/flxwd/78931-big_data_universe_beginning_to_explode
- Department of Defense. (2012). *Department of Defense cloud computing strategy*. Retrieved from <http://dodcio.defense.gov/Portals/0/Documents/Cloud/DOD%20Cloud%20Computing%20Strategy%20Final%20with%20Memo%20%20July%205%202012.pdf>
- Department of the Navy. (n.d.-a). Chief of Naval Personnel – Responsibilities [Information on page]. Retrieved January 6, 2016, from http://www.navy.mil/navydata/leadership/cnp_resp.asp
- Department of the Navy. (n.d.-b). Navy Enlisted System [Information on page]. Retrieved January 6, 2016, from <http://www.public.navy.mil/bupers-npc/organization/npc/IM/corporatessystems/Pages/NEWMainPage.aspx>
- Department of the Navy. (n.d.-c). OPINS [Information on page]. Retrieved January 6, 2016, from <http://www.public.navy.mil/bupers-npc/organization/npc/IM/corporatessystems/Pages/opins.aspx>
- Department of the Navy. (2015) Talent Management initiatives [Information on page]. Retrieved from <http://www.secnav.navy.mil/innovation/Documents/2015/05/TalentManagementInitiatives.pdf>
- Grover, M., Malaska, T., Seidman, J., Shapira, G. (2015). *Hadoop application architectures*. Sebastopol, CA: O'Reilly Media, Inc.
- Hall, T. S. (2006). CNP shares strategic vision while visiting NAS Pensacola [Newspaper story]. Retrieved from http://www.navy.mil/submit/display.asp?story_id=24088

- Hamilton, B.A. (2015). Intelligent Workbook (IW) user manual: Navy Manpower Programming and Budget System (NMPBS)/OPNAV N10. Retrieved from <https://nmpbs.n10.npc.navy.mil>
- Hecht, R. & Jablonski, S. (2011). NoSQL evaluation: A use case oriented survey. *2011 International Conference on Cloud and Service Computing*, 336–341. doi: 10.1109/CSC.2011.6138544
- Henschen, D. (2013, April). Cloudera Impala brings SQL querying to Hadoop. *Information Week*. Retrieved from <http://www.informationweek.com/software/information-management/cloudera-impala-brings-sql-querying-to-hadoop/d/d-id/1109745?>
- Hoffman, S. (2013). *Apache Flume: Distributed log collection for Hadoop*. Birmingham, UK: Packt Publishing Ltd.
- Hyperion. (2000). *The role of the OLAP server in a data warehouse solution* [Information on page]. Retrieved from <http://www.oracle.com/technetwork/middleware/bi-foundation/olap-in-a-data-warehousing-solution-128690.pdf>
- IBM. (n.d.). What is MapReduce [Information on page]. Retrieved January 2016, from <https://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/>
- IBM. (2010). *Introduction to IBM SPSS modeler and data mining: Student guide*. Retrieved from <http://www.exitcertified.com/training/ibm/spss/modeler/introduction-data-mining-7957-detail.html>
- Jacobs, A. (2009). The pathologies of big data. *Communications of the ACM*, 52(8), 36–44. doi: 10.1145/1536616.1536632
- Jing, H., Haihong, E., Guan, L., & Jian, D. (2011). Survey on NoSQL database. *2011 6th International Conference on Pervasive Computing and Applications (ICPCA)*, 363-366. doi:10.1109/ICPCA.2011.6106531
- Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big data: Issues and challenges moving forward. *2013 46th Hawaii International Conference on System Sciences (HICSS)*, 995-1004. doi:10.1109/HICSS.2013.645.
- Karant, S. (2014). *Mastering Hadoop*. Birmingham, UK: Packt Publishing.
- Kroenke, D. M. & Auer, D. J. (2014). Big data, data warehouses, and business intelligence systems. *Database processing: Fundamentals, design, and implementation* (13th ed.) (pp. 534–579). New Jersey: Pearson.

- LaGrone, S. (2014, December 2). Interview: U.S. Navy Personnel Chief worries over potential service retention problems. *U.S. Naval Institute News*. Retrieved from <http://news.usni.org/2014/12/02/interview-u-s-navy-personnel-chief-worries-potential-service-retention-problems>
- LeBlanc, P., Moss, J. M., Sarka, D., & Ryan, D. (2015). *Applied Microsoft business intelligence*. Indianapolis, IN: John Wiley & Sons, Inc.
- Linstedt, D. (2015). Data vault basics [Information on page]. Retrieved from <http://danlinstedt.com/solutions-2/data-vault-basics/>
- Loh, W.Y., & Shih, Y. (1997). Split selection methods for classification trees. *Statistica Sinica*, 7(4), 815-840. Retrieved from <http://www3.stat.sinica.edu.tw/statistica/oldpdf/A7n41.pdf>
- Loshin, D. (2012). *Business intelligence* (2nd ed.). Waltham, MA: Morgan Kaufmann.
- Mabus, R. (2015, May). Department of the Navy talent management address to the brigade of midshipmen. Presented at U.S. Naval Academy, Annapolis, MD. Retrieved from http://navylive.dodlive.mil/files/2015/05/USNA-Talent-Management-Speech_FINAL.pdf
- Nava, J., & Hernández, P. (2012). Optimization of a hybrid methodology (CRISP-DM). In M. Khosrow-Pour (Ed.), *Data mining: Concepts, methodologies, tools, and applications* (pp. 1998–2020). Hershey, PA: IGI Global.
- Navy Manpower Analysis Center. (2015). *NAVMAC: At a glance brief* [PowerPoint slides]. Retrieved from [http://www.public.navy.mil/bupers-npc/organization/navmac/Pages/NAVMAC Information.aspx](http://www.public.navy.mil/bupers-npc/organization/navmac/Pages/NAVMAC%20Information.aspx)
- Navy Personnel Command. (2013). 2020 Strategic vision 2013 update. Retrieved from <http://www.public.navy.mil/bupers-npc/organization/npc/spio/Documents/2020%20Vision%20FINAL%20041013.pdf>
- Office of the Chief of Naval Operations. (2012a, May 14). *Mission, functions, and tasks of the Naval Education and Training Command* (OPNAV Instruction 5450.336C). Washington, D.C.: Chief of Naval Operations.
- Office of the Chief of Naval Operations. (2012b, Jul 24). *Missions and functions of Bureau of Naval Personnel* (OPNAV Instruction 5430.47E). Washington, D.C.: Chief of Naval Operations.
- Office of the Chief of Naval Operations. (2015, Jun 24). *Navy total force manpower policies and procedures* (OPNAV Instruction 1000.16L). Washington, D.C.: Chief of Naval Operations.

- Office of the Chief of Naval Operations N13. (2015, Nov 3). *Manual of Navy Officer manpower and personnel classifications Volume I: Major code structures*. (NAVPERS 15839I). Washington, D.C: Chief of Naval Operations N13.
- Office of the Chief of Naval Personnel. (2015). *Navy Manpower Programming and Budget System (NMPBS): Overview* [PowerPoint Slides]. Retrieved from <https://nmpbs.n10.npc.navy.mil>
- Oliver, A. C. (2014, December). Storm or Spark: Choose your real-time weapon. *InfoWorld*. Retrieved from <http://www.infoworld.com/article/2854894/application-development/spark-and-storm-for-real-time-computation.html>
- Olsen D. L., & Delen, D. (2008). *Advanced data mining techniques*. Berlin: Springer-Verlag.
- Padhy, R. P., Patra, M. R., & Satapathy, S. C. (2011). RDBMS to NoSQL: Reviewing some next-generation non-relational database's. *International Journal of Advanced Engineering Sciences and Technologies*, 11(11), 15–30. Retrieved from http://www.researchgate.net/publication/265062016_RDBMS_to_NoSQL_Reviewing_Some_Next-Generation_Non-Relational_Database's
- PMW 240. (2015a) Navy Standard Integrated Personnel System [Information on page]. Retrieved from http://www.public.navy.mil/spawar/PEOEIS/SWP/Documents/FactSheets/FS_NS_IPS.pdf
- PMW 240. (2015b). Navy Training Management and Planning System [Information on page]. Retrieved from http://www.public.navy.mil/spawar/PEOEIS/SWP/Documents/FactSheets/FS_NT_MPS.pdf
- Pokorny, J. (2013). NoSQL databases: A step to database scalability in web environment. *International Journal of Web Information Systems*, 9(1), 69–82. doi:10.1108/17440081311316398
- Russell, J. (2014). *Getting started with Impala*. Sebastopol, CA: O'Reilly Media, Inc.
- Sammer, E. (2012). *Hadoop operations*. Sebastopol, CA: O'Reilly Media, Inc.
- Sawant, N., & Shah, H. (2013). *Big data application architecture Q&A: A problem - solution approach* (1st ed.). Berkeley, CA: Apress.
- Shreedharan, H. (2014). *Using Flume*. Sebastopol, CA: O'Reilly Media Inc.
- Shripav, S. (2014). *Learning Hbase*. Birmingham, UK: Packt Publishing.

- Shukla, V. (2013). Hadoop security: Today and tomorrow [Information on page]. Retrieved from <http://hortonworks.com/blog/hadoop-security-today-and-tomorrow/>
- Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1–10. doi: 10.1109/MSST.2010.5496972
- Singh, K., & Kaur, R. (2014). Hadoop: Addressing challenges of big data. *2014 IEEE International Advance Computing Conference (IACC)*, 686–689. doi:10.1109/IAdCC.2014.6779407.
- Strauch, C. & Kriha, W. (2011). NoSQL databases. *Lecture Notes, Stuttgart Media University*. Retrieved from <http://webpages.uncc.edu/xwu/5160/nosql dbs.pdf>
- Syed, A. R., Gillela, K., & Venugopal, C. (2013). The future revolution on big data. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(6). Retrieved from <http://www.ijarcce.com/upload/2013/june/44-Abdul%20Raheem-The%20Future%20Revolution%20on%20Big%20Data.pdf>
- Teller, S. (2015). *Hadoop essentials*. Birmingham, UK: Packt Publishing.
- Tiwary, C. (2015). *Learning Apache Mahout*. Birmingham, UK: Packt Publishing.
- Vaisman, A., & Zimányi, E. (2014). Data warehouse concepts. *Data warehouse systems* (pp. 53–87) Berlin: Springer.
- Virtual Hadoop [Information on page]. (2013). Retrieved from <http://wiki.apache.org/hadoop/Virtual%20Hadoop>
- Wadkar, S., & Siddalingaiah, M. (2014). *Pro Apache Hadoop* (2nd ed.). Berkeley, CA: Apress.
- White, T. (2015). *Hadoop: The definitive guide* (4th ed.). Sebastopol, CA: O'Reilly Media.
- Withanawasam, J. (2015). *Apache Mahout essentials*. Birmingham, UK: Packt Publishing.
- Zikopoulos, P. C., Eaton, C., deRoos, D., Deutsch, T., & Lapis, G. (2012). *Understanding big data: Analytics for enterprise class Hadoop and streaming data*. New York: McGraw-Hill.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California